

Advanced Monero Wallet Forensics

Demystifying Off-Chain Artifacts to Trace Privacy-Preserving Cryptocurrency Transactions

Jeongin Lee, Geunyeong Choi, Jihyo Han, Jungheum Park



KOREA UNIVERSITY
School of Cybersecurity



DFRC Korea University
Digital Forensic Research Center

Table of Contents

- ❑ **Introduction and Background**
- ❑ **Components of Monero Wallet and Research Scope**
- ❑ **Off-chain Artifacts of Monero Wallet**
- ❑ **Conclusion and Future Directions**

Keywords



Digital Forensics

Live Forensics

Memory Forensics

Cryptocurrency

Transaction Tracing

Monero

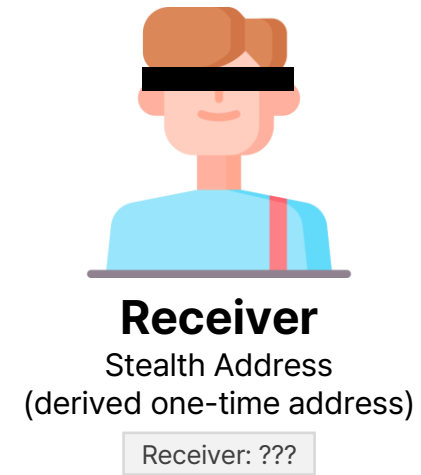
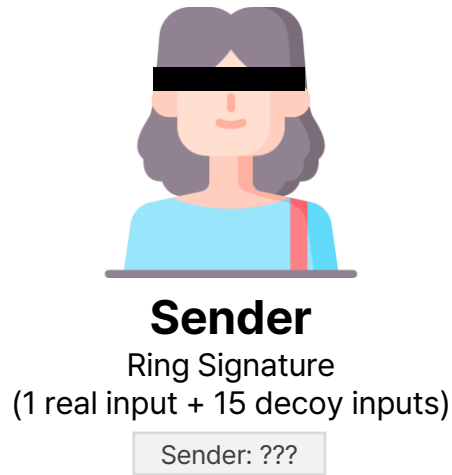
Forensic Tool Development

Introduction and Background

Introduction and Background

Research Motivation

- **Cryptocurrency's Anonymity and Traceability**
 - **Disconnection between wallet owners and addresses** makes tracing difficult
- **Key Characteristics of Monero**



- **Forensic Challenges and Research Gaps**
 - Existing studies lack integrated approaches for memory-file analysis and key decryption

Introduction and Background

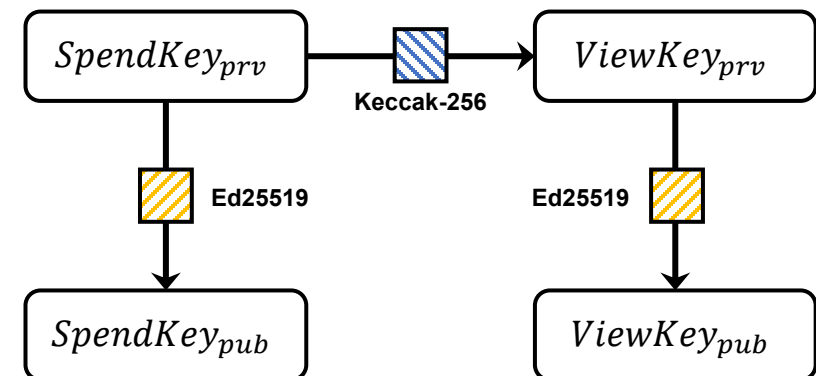
Monero Wallet Keys

- **SpendKey_{prv}**

→ Enables full control of all wallet funds, key recovery, and wallet restoration

Key Type	Function	Risk if Exposed
SpendKey _{pub}	Used to generate the Monero address	-
SpendKey _{prv}	Allows spending funds	Funds can be stolen
ViewKey _{pub}	Used to generate the Monero address	-
ViewKey _{prv}	Allows viewing transaction history	Transaction history may be exposed (but funds remain safe)

- **All wallet keys → derived from the SpendKey_{prv}**



Components of Monero Wallet and Research Scope

Components of Monero Wallet and Research Scope

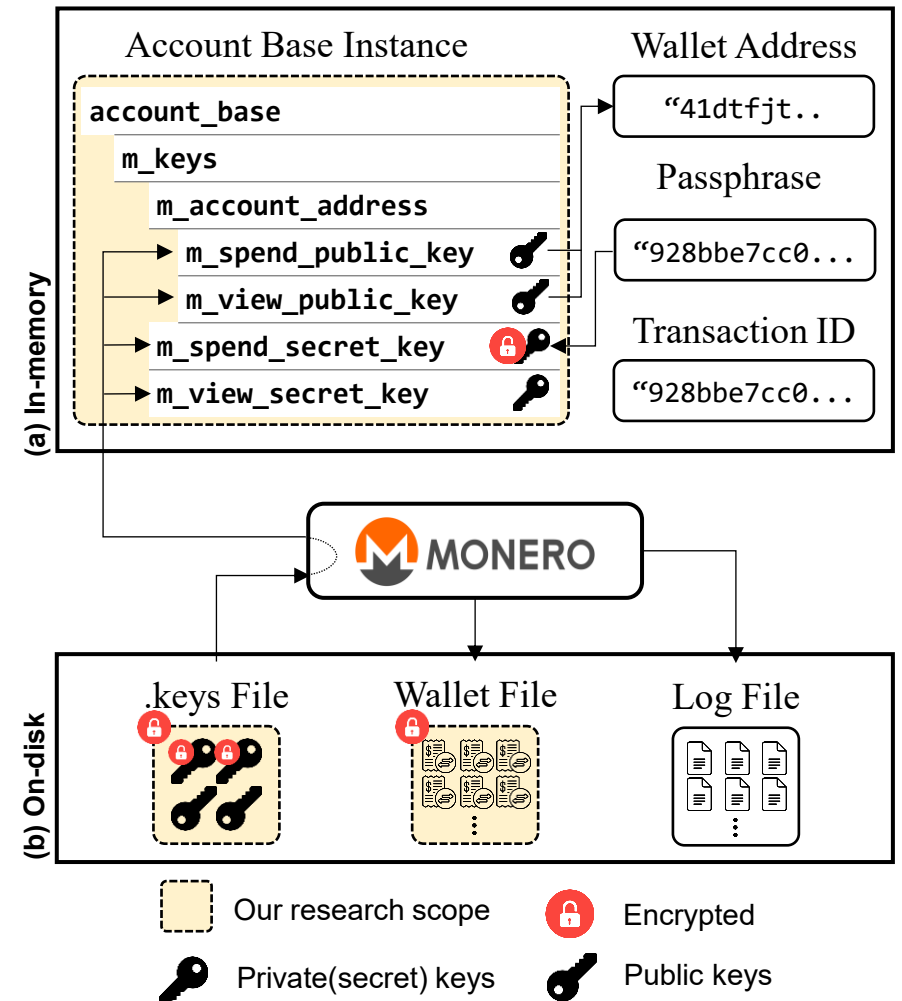
Elements of Monero Wallets

■ In-memory

- Structure consisting of four Monero keys managed by the **account_base** instance

■ On-disk

- Storage of encrypted keys and wallet files, containing all key materials and transaction data
- Wallet keys file : stores the four keys
- Wallet cache file : caches transaction data



Components of Monero Wallet and Research Scope

Research Scope and Questions

RQ1. How can raw byte stream components, such as an `account_base` instance of both Monero CLI and Monero GUI, be identified and decrypted from a memory dump?

RQ2. How can the wallet keys and cache file be decrypted and meaningfully interpreted?

RQ3. Can our proposed approach be applied to extract public and private keys from other cryptocurrency wallet applications that support Monero?

Off-chain Artifacts of Monero Wallet

Off-chain Artifacts of Monero Wallet

Understanding of account_base Class

■ Memory management of SpendKey_{pub}, ViewKey_{pub}, SpendKey_{prv}, ViewKey_{prv}

```
namespace cryptonote {
    class account_base {
    private:
        struct account_keys m_keys {
            struct account_public_address m_account_address {
                crypto::public_key m_spend_public_key;
                crypto::public_key m_view_public_key;
            }
            crypt::secret_key m_spend_secret_key;
            crypt::secret_key m_view_secret_key;
            std::vector<crypto::secret_key> m_multisig_keys;
            hw::device *m_device = &hw::get_device("default");
            crypto::chacha_iv m_encryption_iv;
        } // end of account_keys

        uint64_t m_creation_timestamp;
    } // end of account_base

} // end of cryptonote namespace
```

Variable name	Offset	Size	Type	Description
m_spend_public_key	0	32	uint8[32]	SpendKey _{pub} in plaintext
m_view_public_key	32	32	uint8[32]	ViewKey _{pub} in plaintext
m_spend_secret_key	64	32	uint8[32]	Encrypted SpendKey_{prv}
m_view_secret_key	96	32	uint8[32]	ViewKey _{prv} in plaintext
m_multisig_keys	128	24	std::vector	Keys for multisignature
m_device	152	8	hw::device*	External device info for key storage
m_encryption_iv	160	8	uint8[8]	IV value used to encrypt the SpendKey _{prv}
m_creation_timestamp	168	8	uint64	Timestamp the wallet was created (UNIX Seconds)

Off-chain Artifacts of Monero Wallet

Understanding of account_base Class

account_base found in memory

Real spend and view keys (public and private)

PUBLIC	SPEND	918B12218414B50BF2E145CA944AD758AD6235A837446FF441922B03BA1071AA	✓
	VIEW	0F3B9E541E3A58735CB5276B5E494920BC92BCFB37698CC18300A8E1C402DE2A	✓
PRIVATE	SPEND	2148ACA02201F035FEED1EC7B95C869F6E4A43D8ECD2070ECC262C0B652CDC0B	≠
	VIEW	57BF9F82AE9C36AAE240B9155971194E88FE3429AEB6293F9E0FF4A6DFDD0000	✓

Not equal

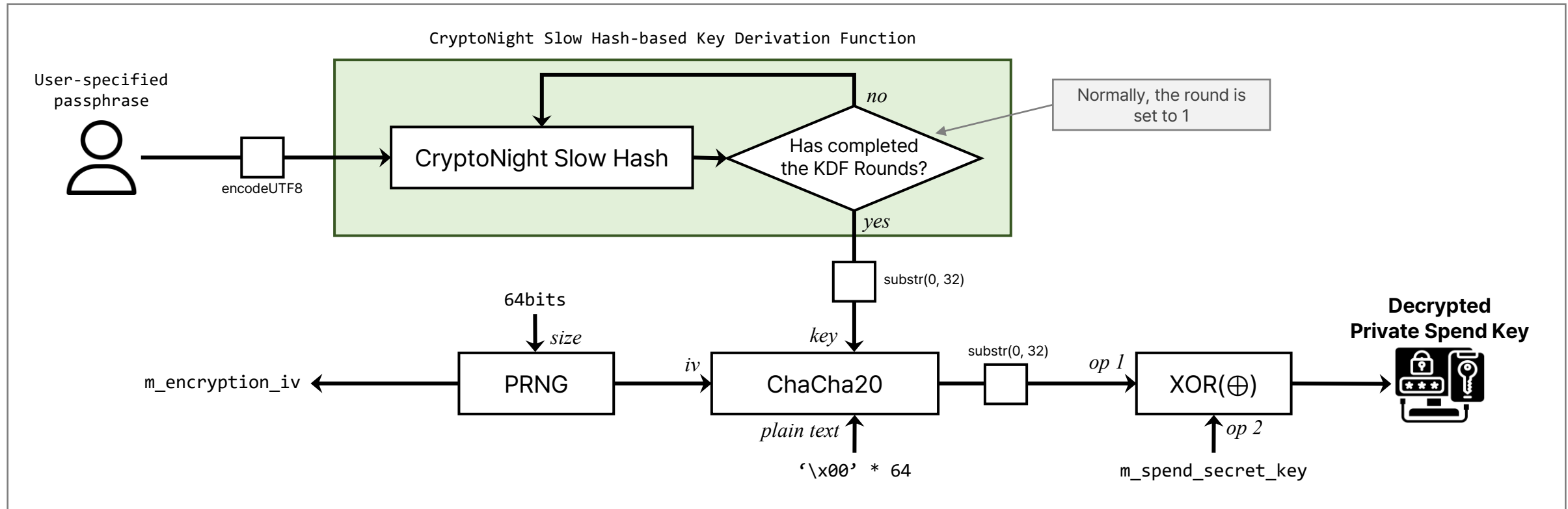
Fields of 'account_base' class

		Address	Offset	Data	Decoded
m_spend_public_key	●				' < ! , , . μ . ò á E Ê ” J × X . b ” 7 D o ô A ’ + . º . q ¢ . ; T . : X s \ μ ' k ^ I I . ¼ ’ % 0 7 i E Á f . ” á Ä . þ * ? % 0 (^ V ä U L . + ß “ þ 3 æ & 7 ¢ ä 0 .) t F . l Î T < . Ó W ¿ ÿ , ¢ æ 6 ¢ ä @ º . Y q . N ^ þ 4) ¢ Ÿ) ? ž . ô ! B Ý Ä .. - × ... x ¢ ä , d z C ¢ g ø } g . . .
m_view_public_key	■	0x102C`1C20	0x00	91 8B 12 21 84 14 B5 0B F2 E1 45 CA 94 4A D7 58	
m_spend_secret_key	▲	0x102C`1C30	0x10	AD 62 35 A8 37 44 6F F4 41 92 2B 03 BA 10 71 AA	
m_view_secret_key	◆	0x102C`1C40	0x20	0F 3B 9E 54 1E 3A 58 73 5C B5 27 6B 5E 49 49 20	
		0x102C`1C50	0x30	BC 92 BC FB 37 69 8C C1 83 00 A8 E1 C4 02 DE 2A	
		0x102C`1C60	0x40	3F 89 F0 28 88 56 E3 55 4C 10 2B DF 93 87 33 E6	
m_multisig_keys	■	0x102C`1C70	0x50	26 37 AE E2 30 2E 29 74 46 0A 6C CE 54 3C 9D D3	
		0x102C`1C80	0x60	57 BF 9F 82 AE 9C 36 AA E2 40 B9 15 59 71 19 4E	
m_device	■	0x102C`1C90	0x70	88 FE 34 29 AE B6 29 3F 9E 0F F4 A6 DF DD 00 00	
		0x102C`1CA0	0x80	00 00 00 00 00 00 00 00 00 00 00 00 00 00	
m_encryption_iv	★	0x102C`1CB0	0x90	00 00 00 00 00 00 00 00 C0 0A 06 97 D7 02 00 00	
		0x102C`1CC0	0xA0	A4 AA E2 2C 64 7A 43 AE 67 F8 7D 67 00 00 00 00	
m_creation_timestamp	■				

Off-chain Artifacts of Monero Wallet

Understanding of account_base Class

■ Decryption process of SpendKey_{prv}



Off-chain Artifacts of Monero Wallet

Understanding of account_base Class

■ Method to discover account_base instance

Algorithm: Discovering *account_base* object

```
1 MEMORY_DUMP ← memory dump file stream
2 CPU_ARCH ∈ {4,8}
3
4 procedure discover_account_base_objects(passphrase)
5   results ← []
6   for offset = 0 to sizeof(MEMORY_DUMP)-sizeof(account_base) step CPU_ARCH do
7     data ← read(MEMORY_DUMP, offset, sizeof(account_base))
8     obj ← convert_as(data, typeof(account_base))
9
10    if passphrase ≠ NULL then
11      decSSK ← decrypt_spend_secret_key(passphrase, obj.m_encryption_iv, obj.m_spend_secret_key)
12      if derive_public_key(decSSK) ≠ obj.m_spend_public_key then
13        continue
14      end if
15    end if
16
17    if derive_public_key(obj.m_view_secret_key) ≠ obj.m_view_public_key then
18      continue
19    end if
20
21    if obj.m_creation_timestamp < unix_timestamp('2014-04-18') then
22      continue
23    end if
24    add(results, obj)
25  end for
26  return results
27 end procedure
```

1 Data Transformation

- Scan the memory dump by 4 or 8 bytes and convert it into an 'account_base' object

2 Spend key Verification

- If a **wallet passphrase is provided**, decrypt the $\text{SpendKey}_{\text{prv}}$ and **verify the derived public key**
- Monero uses **ED25519**, so a matching derived public key confirms a valid private key

3 View key Verification

- Verify that the $\text{ViewKey}_{\text{prv}}$ derives to the $\text{ViewKey}_{\text{pub}}$

4 Timestamp Verification

- Verify if the timestamp is after Monero's release date, April 18, 2014

Off-chain Artifacts of Monero Wallet

Discovering Monero's Keys in System Memory

- MoeyEx: A Volatility3 plugin for extracting Monero's account_base instances

F:\volatility3>python vol.py moeyex --file-path="F:\experiment\images\monero_wallet.mem" --passphrase=Monero13

Volatility 3 Framework 2.26.2

Progress: 100.00 PDB scanning finished

Key Value

Public Spend Key 643583978871a389fed7f66ec9afe4160a7d063f8b3bcc9896b668034da7d503

Public View Key ff8cc50a5fce8c842b7a974ad64c80a08aa5766b6ada6996b1d878ecda04aa46

Private Spend Key 90f700ce6831427fd956ebabb8645caf5560a7f21f81b09ef5a73e69f289260c

Private View Key b6fc8bf158663bf06b73cf5978d5f552d338514ade7250c86cc530b77c4a5f0f

Creation Time 2025-05-07 23:08:30

Option

Result

Off-chain Artifacts of Monero Wallet

Coverage of Our Work

- **Monero-supported wallet account_base instance identification**
 - **'Feather'** memory dump reveals Monero account structure

Feather's wallet keys

Keys

Restore height
3463767 (2025-07-27)

Primary address
43vg9Y5zVHDbJuHafX546meV6XkbACsbdc1IXmBKg6aM8YNfxYPNDEw21NyECIhx9gR4kTE1JRMfmigkKV7rmWcEJbMFczP

Secret spend key
b65d2e715665032935f4dc1ab4eaf219c147f9bb360989bfae56b34b98dd6303

Secret view key
721f61fc8ca7bbb360cc187baf6e499d0228322fcc048d3e2db04795675cda0e

Public spend key
3cc4ee049fb9a4cd1e34d670e87b3ae01ac716b8e82628d1533c2ee7fa6bba2d

Public view key
159111c4a881a0060484fbd70906a38fdf82af61831ca8f938569245c1f6bf9b

'MoeyEx' execution result

F:\volatility3>python vol.py moeyex --file-path="F:\experiment\images\feather.mem" --passphrase=feather

Volatility 3 Framework 2.26.2

Progress: 100.00 PDB scanning finished

Key	Value
Public Spend Key	3cc4ee049fb9a4cd1e34d670e87b3ae01ac716b8e82628d1533c2ee7fa6bba2d
Public View Key	159111c4a881a0060484fbd70906a38fdf82af61831ca8f938569245c1f6bf9b
Private Spend Key	b65d2e715665032935f4dc1ab4eaf219c147f9bb360989bfae56b34b98dd6303
Private View Key	721f61fc8ca7bbb360cc187baf6e499d0228322fcc048d3e2db04795675cda0e
Creation Time	2014-06-07 15:00:00

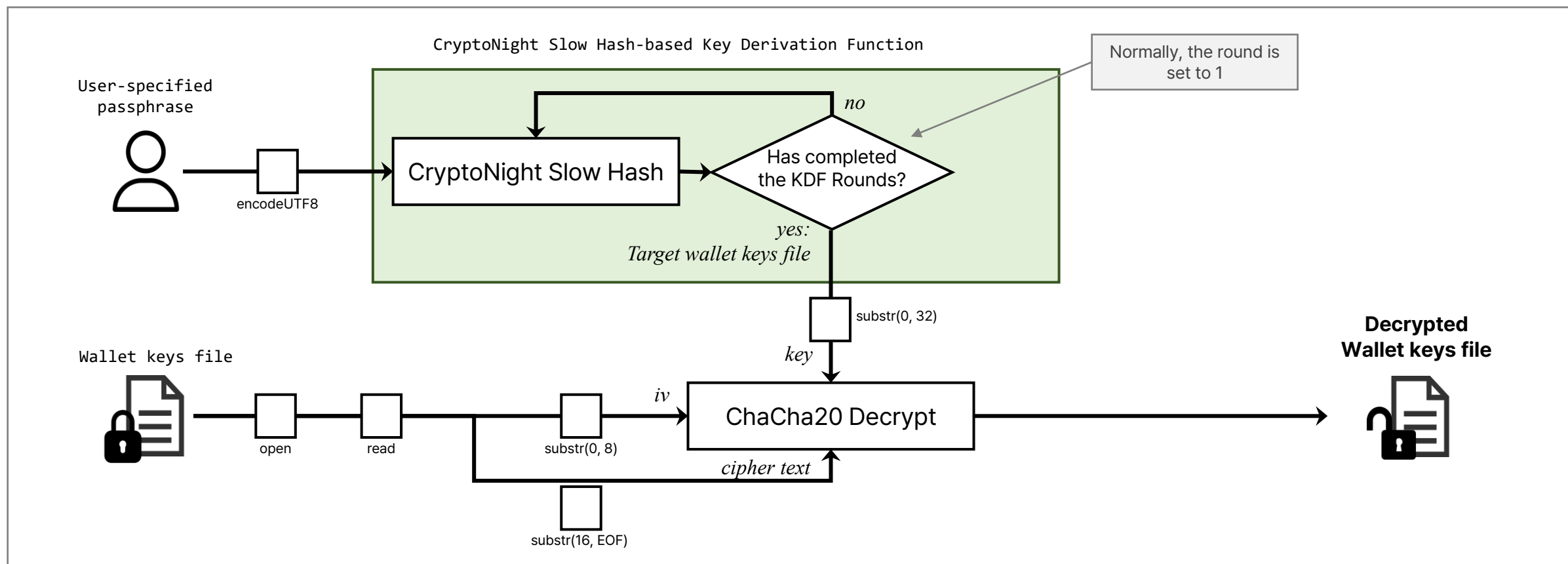
Fixed value

Off-chain Artifacts of Monero Wallet

Non-volatile Data: Wallet Keys File Internals

■ Diagram of wallet keys file decryption process

- **iv** : The first 8 bytes of the file starting from offset 0



Off-chain Artifacts of Monero Wallet

Non-volatile Data: Wallet Keys File Internals

- Resulting output is in JavaScript Object Notation (JSON) format

Decrypted Data (Formatted JSON):

```
{
  "key_data": "\u0001\u0011\u0001\u0001\u0001\u0001\u0001\u0002\u0001\u0001\u0014m_creation_timestamp\u0005\u0019g\u0000\u0000\u0000\u0000\u0006m_keys\u0010\u0011m_account_address\u0012m_spend_public_key\u0016m_spend_secret_key\u0017m_view_public_key\u0018m_view_secret_key\u0019m_encryption_iv\u001am_spend_secret_key\u001bm_view_secret_key\u001cm_tx_cayno.sw\u0014",
  "seed_language": "English",
  "key_on_device": 0,
  "watch_only": 0,
  "multisig": 0,
  "multisig_threshold": 0,
  "always_confirm_transfers": 1,
  "print_ring_members": 0,
  "store_tx_info": 1,
  "default_mixin": 0,
  "default_priority": 0,
  "auto_refresh": 1,
  "refresh_type": 1,
  "refresh_height": 3357451,
  "skip_to_height": 0,
  "confirm_non_default_ring_size": 1,
  "ask_password": 2,
  "max_reorg_depth": 100,
  "min_output_count": 0,
  "min_output_value": 0,
  "default_decimal_point": 12,
  "merge_destinations": 0,
```



Key data blob

```
"min_output_value": 0,
"default_decimal_point": 12,
"merge_destinations": 0,
"confirm_backlog": 1,
"confirm_backlog_threshold": 0,
"confirm_export_overwrite": 1,
"auto_low_priority": 1,
"nettype": 0,
"segregate_pre_fork_outputs": 1,
"key_reuse_mitigation2": 1,
"segregation_height": 0,
"ignore_fractional_outputs": 1,
"ignore_outputs_above": 18446744073709551615,
"ignore_outputs_below": 0,
"track_uses": 0,
"background_sync_type": 0,
"show_wallet_name_when_locked": 0,
"inactivity_lock_timeout": 90,
"setup_background_mining": 2,
"subaddress_lookahead_major": 50,
"subaddress_lookahead_minor": 200,
"original_keys_available": 0,
"export_format": 0,
"load_deprecated_formats": 0,
"encrypted_secret_keys": 1,
"device_name": "",
"device_derivation_path": "",
"persistent_rpc_client_id": 0,
"auto_mine_for_rpc_payment": 0.0,
"credits_target": 0,
"enable_multisig": 0
}
```

Off-chain Artifacts of Monero Wallet

Non-volatile Data: Wallet Keys File Internals

■ Portable Storage (PS) format

Fields of 'key_data'

header(signature 8bytes + version 1bytes).....

section(entry(name-value pair) count)

entry_1 'm_creation_timestamp'

entry_2 'm_keys'

entry_3 'm_account_address'

entry_4 'm_spend_public_key'

entry_5 'm_view_public_key'

entry_6 'm_encryption_iv'

entry_7 'm_spend_secret_key'

entry_8 'm_view_secret_key'

'key_data' object in wallet keys file

Offset	Data	Decoded
0x000	01 11 01 01 01 01 02 01 01 08 14 6D 5F 63 72 65	m_cre
0x010	61 74 69 6F 6E 5F 74 69 6D 65 73 74 61 6D 70 05	ation_timestamp.
0x020	29 F4 01 68 00 00 00 00 06 6D 5F 6B 65 79 73 0C	)ô.h....m_keys.
0x030	10 11 6D 5F 61 63 63 6F 75 6E 74 5D 61 64 64 72	.. m_account_addr
0x040	65 73 73 0C 08 12 6D 5F 73 70 65 6E 64 5F 70 75	ess...m_spend pu
0x050	62 6C 69 63 5F 6B 65 79 0A 80 32 46 B6 5A 87 F1	blic_key.€2F¶Z†ñ
0x060	4E 4C 1E 08 AC 02 CD AC FB 02 B4 3D 86 85 58 A3	NL...î-û.'=+...Xf
0x070	40 4C 4F 90 F4 26 13 8C 4E ED 11 6D 5F 76 69 65	@LO.ô&.€Ní.m_vie
0x080	77 5F 70 75 62 6C 69 63 5F 6B 65 79 0A 80 24 B7	w_public_key.€\$.
0x090	14 6B E8 6D 34 7F 54 DF B0 10 AB 91 DB 7B B0 71	.Kèm4.Tß°.«'Û{°q
0x0A0	3C 3B EA 76 34 48 63 12 2E ED E4 2F B5 03 0F 6D	<;êv4Hc..íä/µ..m
0x0B0	5F 65 6E 63 72 79 70 74 69 6F 6E 5F 69 76 0A 20	_encryption_iv.
0x0C0	CE 0A FC CD 8F AB 5A AA 12 6D 5F 73 70 65 6E 64	Î.üí.«Zª.m_spend
0x0D0	5F 73 65 63 72 65 74 5F 6B 65 79 0A 80 2B 5C EC	_secret_key.€+\ì
0x0E0	6B 59 8C 60 80 66 08 DB 0A 12 47 E9 5B F7 D0 3A	kY€`€f.Ö..Gé[÷ð:
0x0F0	6B E8 49 8F C6 8B 0F BF 6D E9 B1 36 B0 11 6D 5F	kèI.Æ<.;mé±6°.m_
0x100	76 69 65 77 5F 73 65 63 72 65 74 5F 6B 65 79 0A	view_secret_key.
0x110	80 D4 3E A4 BC 68 BB F1 B4 09 59 AF 7A 49 59 77	€Ô>¤%h»ñ'.Y`zIYw
0x120	16 0D 3E 2F 4C A2 0C 86 73 CF 22 39 0C 6D 52 30	...>/L\$.†sİ"9ÀmR0
0x130	C8	È

https://github.com/monero-project/monero/blob/master/docs/PORTABLE_STORAGE.md

Off-chain Artifacts of Monero Wallet

Non-volatile Data: Wallet Keys File Internals

- An example output of wallet keys file decryption script

F:\>python decrypt_keys_file.py --file="F:\experiment\images\tx_test\tx_test_3.keys" --password="txpwd3" → **Option**

Result

```
{
  "key_data": {
    "creation timestamp": "2025-05-14 08:02:58",
    "spend public key": "71ecf07d48aacf986a012805f780bc952c1f6b327c24907d2f764c226c44fb35",
    "view public key": "fb3971aa320573d97d3ba2bf65bcc261cc560a1b3794fe8c8fbfa6a2d238d8d2",
    "encryption iv": "1e0b807649e25096",
    "spend secret key": "78c03f715455e9defc933bc25f6bca6928af67b127a1f10b746fb518fb658c08",
    "view secret key": "8c1190c0b7bc6370114f9590bd77e839fbe5c74f3cd5bd569ac29adcc4f78d0b"
  },
  "seed_language": "English",
  ...
}
```

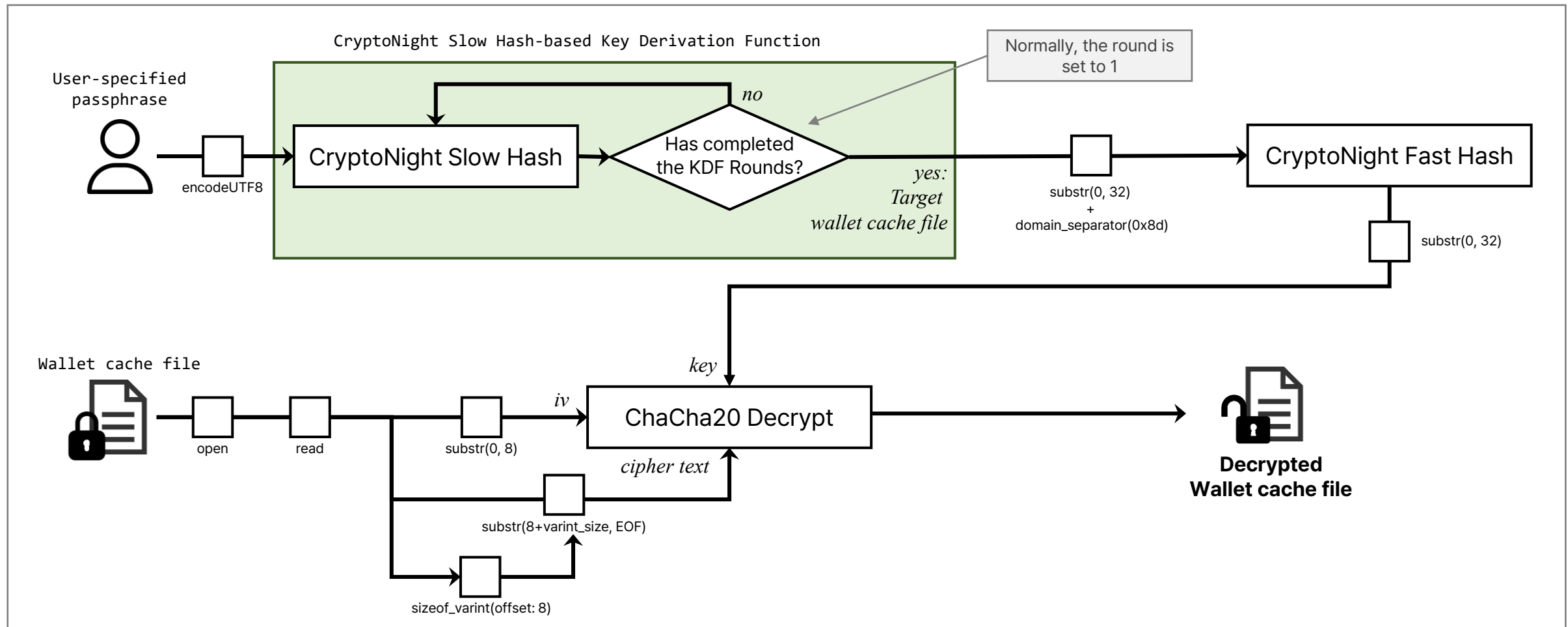
→ **4 Key values**

→ **49 key-value pair data related to wallet settings**

Off-chain Artifacts of Monero Wallet

Non-volatile Data: Wallet Cache File Internals

■ Diagram of wallet cache file decryption process



Off-chain Artifacts of Monero Wallet

Non-volatile Data: Wallet Cache File Internals

- Deserialization of wallet file format

Field Name	Size(bytes)	Description
MAGIC_FIELD("monero wallet cache")	20	Magic string "monero wallet cache"
VERSION_FIELD(2)	1	Cache version (currently 2)
m_blockchain	variable	Blockchain hash list
m_transfers	variable	List of outputs owned
m_account_public_address	64	Public spend/view key
m_key_images	variable	Key image map
m_unconfirmed_txs	variable	Unconfirmed outgoing transaction map
m_payments	variable	Confirmed incoming transaction map
m_tx_keys	variable	Outgoing transaction key map
m_confirmed_txs	variable	Confirmed outgoing transaction map

⋮

Off-chain Artifacts of Monero Wallet

Non-volatile Data: Wallet Cache File Internals

- **Deserialization of wallet file format**

- Incoming transaction-related sub-fields in **m_payments**

Field Name	Size(bytes)	Description
m_tx_hash	32	Transaction hash (tx id)
m_amount	varint(1-10)	Received amount (in piconero)
m_fee	varint(1-10)	Transaction fee
m_block_height	varint(1-10)	Block height
m_timestamp	varint(1-10)	Block creation timestamp

- Outgoing transaction-related sub-fields in **m_confirmed_txs**

Field Name	Size(bytes)	Description
m_tx	variable	Transaction header
m_amount_in	varint(1-10)	Before sending amount
m_amount_out	varint(1-10)	Amount after subtracting fee
m_change	varint(1-10)	Change
m_block_height	varint(1-10)	Block height
m_dests	varialbe	Destination address etc.
m_timestamp	varint(1-10)	Block creation timestamp

Off-chain Artifacts of Monero Wallet

Non-volatile Data: Wallet Cache File Internals

- An example output of wallet cache file decryption script

F:\>python decrypt_cache_file.py --file="F:\experiment\images\tx_test\tx_test" --password="txpwd1" → Option								Result
Transactions Summary								
Type	Timestamp	BlockHeight	TX ID	TX Key		Amount	Fee	
Change	Destination Address		Public Key 1	Public Key 2				
in	2025-05-12 10:10:20	3409961	bb957ca1aa4a548fcb09f1ba70abc5cc90a4f85d15922bb13d40bab96cb66c6b -			0.009969280000	0.000030720000	Tx 1
			-			-	-	
out	2025-05-13 03:46:14	3410487	ad49c70b4e05b9f956a99523068ab9b08229168befe0e2091c97dd83489b3a39			0.009000000000	0.000030660000	Tx 2
	0449270cdb	c22e2250656d6a1fc0efe31b6a6eb4d16e3b3535039e5bbb62260c	0.000938620000					
	47EFR6pgmm1Cx9gUnAVEYdVdi5mcYKjqJPzwPxS8S7srB9eUjtGbJJwHHB8bywaiGKS8MBbyLNwU8X9HXgneRDy9QccCykX							
	71e666a531f2fd8d56eefedcb1e0e6c11508c19d027557ca0ca022c444630743							
	1d98816665225339b7a24d9e3162c586e899f1cf5a8eb6cc16928ed67aea5561							
out	2025-05-14 08:07:48	3411357	ee0f5701dafee5649eeadc4f2c1a7eaa85cdb73828010d5c3ff313882a987bb3			0.000300000000	0.000122960000	Tx 3
	7b0cdab0bdc7d6f0d7af8137468c99bcddb0d6e291b84215cb900de7d2991508					0.000515660000		
	45wWYEdpetNSVcEdMPjLKDRxACs1xy4qMMwTQEWc49dkA2gpxMWJnJEdNvDnNsjoKfHMmNDsNhsWMQWd86nqcX2FQhVUd7m							
	533a0c63b54a0f15b4290529096f8afaac2abc197e053f5ca2ee09c2e6d33a9d							
	3c8cf2ec8d6fa43245668e64326c081329ad3f4cdb565d635bcd9bce816c1342							

Conclusion and Future Directions

Conclusion and Future Directions

Answers to Research Questions

■ RQ1: Memory forensics for Monero

- We can extract Monero keys from memory by scanning for **account_base** objects and decrypting them with a recovered passphrase

■ RQ2: Disk forensics for Monero

- Our method enables complete decryption and analysis of **wallet keys and cache files**, revealing all keys and transaction details

■ RQ3: Coverage of our work

- The same approach works for third-party wallets using Monero CLI code - including **Feather and Monero GUI** - enabling key and transaction recovery across wallets

Conclusion and Future Directions

■ Conclusion

- Identification of Monero CLI key management
 - Based on source code analysis
- Development of MoeyEx scanning tool for memory dump decryption
- Detection of **account_base** instance in wallet memory
 - Applies to Monero GUI and Feather
- Decryption of **wallet keys** and **cache files**
 - For non-volatile storage analysis

■ Future work

- Planned support for multisignature, subaddress, and OS expansion

THANK YOU



web | dfrc.korea.ac.kr

email | jjeongjin@korea.ac.kr