IEEE
SecDev|2021

IEEE COMPUTER SOCIETY

Android Remote Unlocking Service using Synthetic Password: A Hardware Security-preserving Approach

Sungmin Lee, Yoonkyo Jung, Jaehyun Lee, Byoungyoung Lee, and Ted “Taekyoung” Kwon

#IEEESecDev https://secdev.ieee.org/2021

Contents

1. Introduction
2. Background
3. Security by Design
4. Implementation
5. Evaluation
6. Conclusion

1. Introduction

Android Remote Unlocking Service (1/2)

- What is it?: Demo video clip(1 min. 17 sec.)[1]
  - Allows the users unlock their Android device through the Internet

[1] Video clip: https://drive.google.com/file/d/1MUiJLG2GU53x6jQgEagU-VFWowaU_E2q

Android Remote Unlocking Service (2/2)

- Not many manufacturers support for security
  - Remote unlocking service inevitably increases the attack surface
  - Difficult to design and implement a secure remote unlocking service

- Stage changes
  - Users can continue to use the device even after unexpected password forgetting
  - Adopting non-face-to-face services is highly encouraged in the COVID-19 era
  - Android File Based Encryption (FBE) blocks the manufacturer to investigate malfunctions

- Seek to a new remote unlocking service to preserve the security
  - Due to the synthetic password, our design doesn’t require H/W modification

2. Background

Android Security features

- Synthetic Password (SP)
  - In enterprise scenarios, a device user and an owner may be different
  - Device owner of the enterprise scenario should be able to reset the screen lock
  - SP Introduced in Android 8 (or Oreo) using Reset Password Token (RPTkn)
  - DevicePolicyManager (DPM) supports the related APIs based on H/W backed Keystore

- Application sandbox: kernel level app isolation based on UID
  - Android apps cannot communicate directly with each other by default

- Application integrity: developer’s signature isolates each apps

- Application permissions: access controls based on the app signature

3. Security by Design

Design goals (1/2)

- Preserving hardware-backed security
  - Trust anchors must reach to specific Hardware Security Modules (HSMs)
  - RSA private key AES key should not be exposed outside of HSM
  - Even manufacturers cannot unlock the locked device arbitrarily
  - Overlaps multiple security features for poor operation or unexpected mis-implementation

- Two-factor authentication: what-you-know and what-you-have
  - Only the device possessing user can start the remote unlocking service for the device

- Distributed authority: Account, Database(+HSM), and Web(+HSM) servers
  - If attackers tries unlocking an arbitrary device, they must crack all the three servers

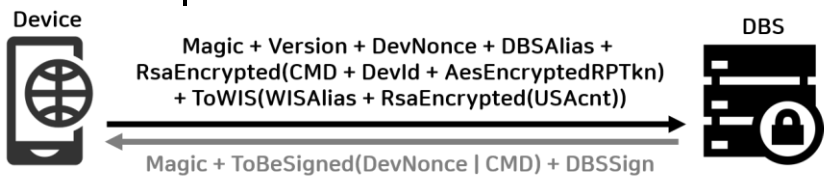
Design goals (2/2)

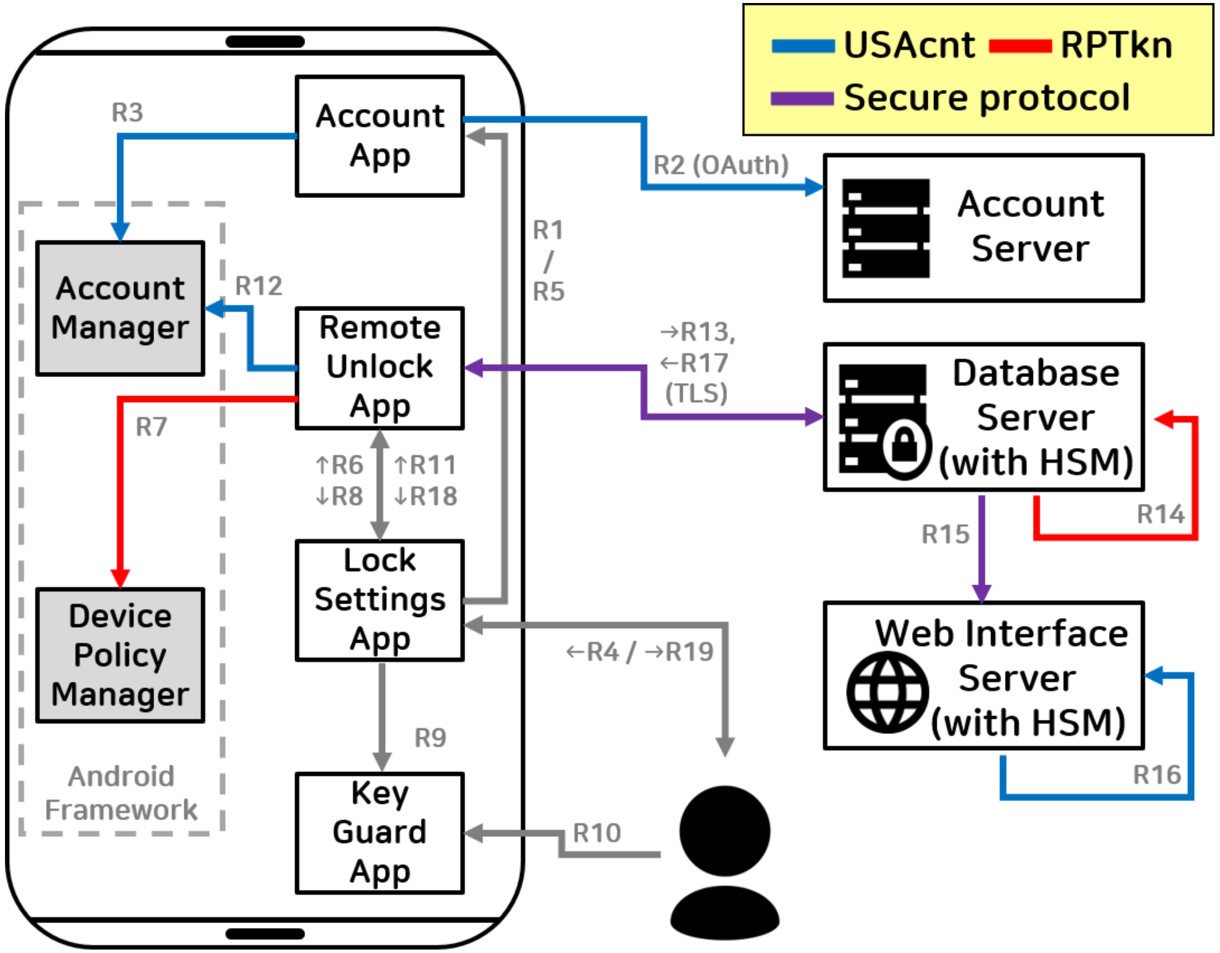
- Trust-boundary minimization: even system app cannot access the RPTkn
  - Platform key for system-level permission is shared with system-privileged app developers
  - We added a new access control (Call-stack monitoring) to the Android permission system

- Key management and compatibility
  - Service administrators should be able to change the public/private key pairs
  - Considers future expansion of the service functionalities

Data Flow Diagrams (1/2)

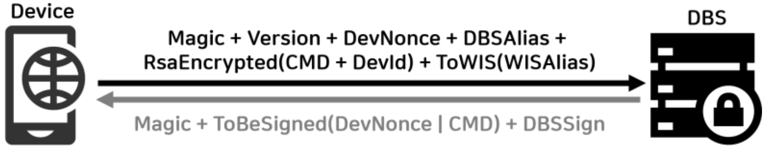
- Major Components
  - Reset password token (RPTkn)
  - User Service Account (USAcnt)
  - Device Identifier (DevId)
  - Remote Unlocking App (RUApp)
  - Database Server (DBS)
  - Web Interface Server (WIS)
- Device registration phase
  - 20 steps in high-level view
  - Secure protocol


Device
DBS
Magic + Version + DevNonce + DBSAlias +
RsaEncrypted(CMD + DevId + AesEncryptedRPTkn)
+ ToWIS(WISAlias + RsaEncrypted(USAcnt))
Magic + ToBeSigned(DevNonce | CMD) + DBSSign

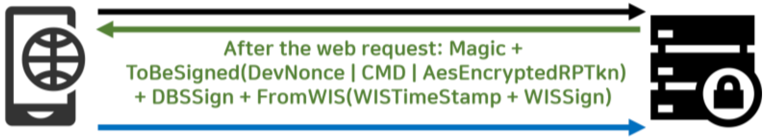


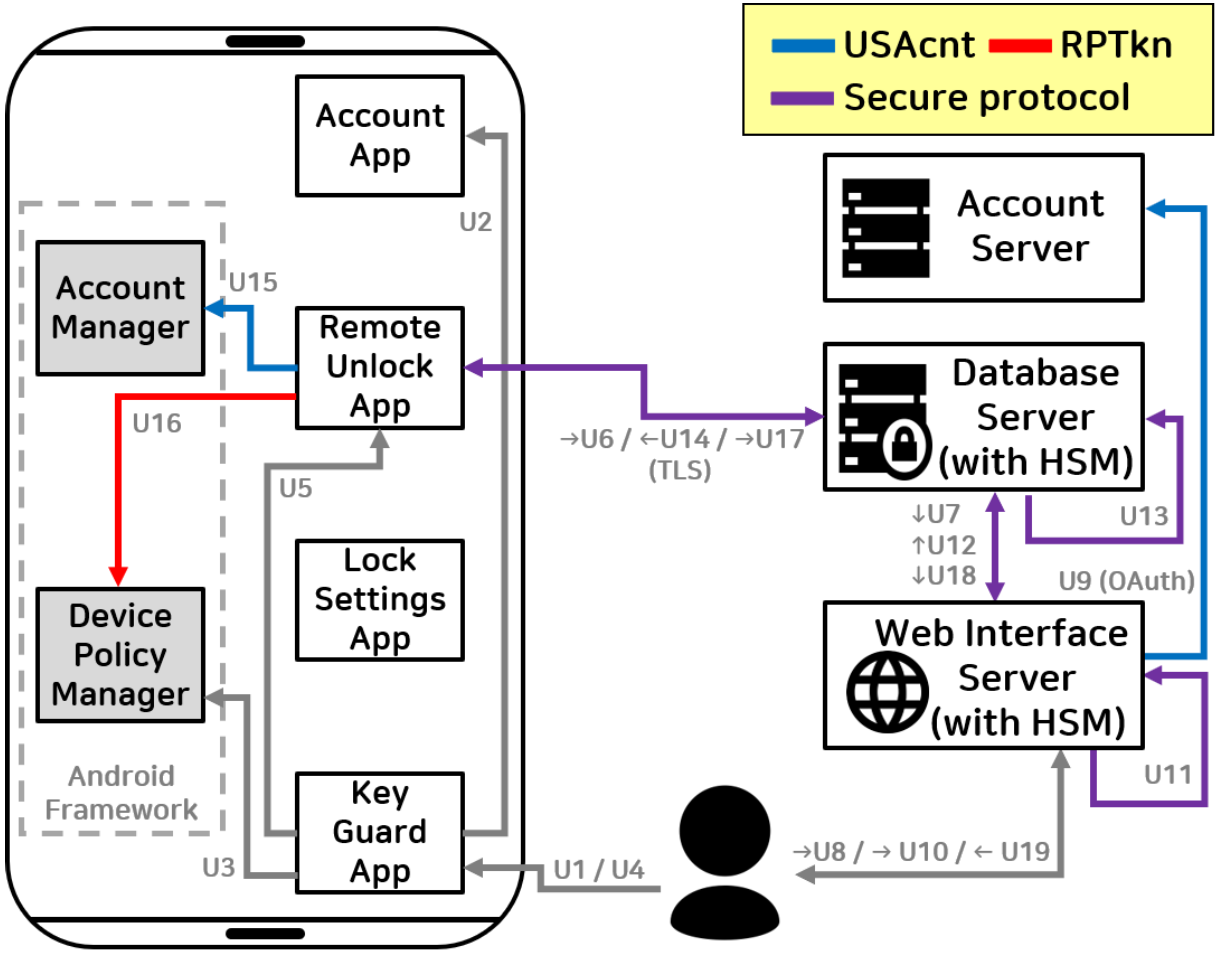
Data Flow Diagrams (2/2)

- Device unlocking phase
  - 19 steps in high-level view
  - Secure protocol
    - Server polling for synchronization
    - Before the user WIS command


Device
DBS
Magic + Version + DevNonce + DBSAlias + RsaEncrypted(CMD + DevId) + ToWIS(WISAlias)
Magic + ToBeSigned(DevNonce | CMD) + DBSSign

    - After the user WIS command


After the web request: Magic + ToBeSigned(DevNonce | CMD | AesEncryptedRPTkn) + DBSSign + FromWIS(WISTimeStamp + WISSign)

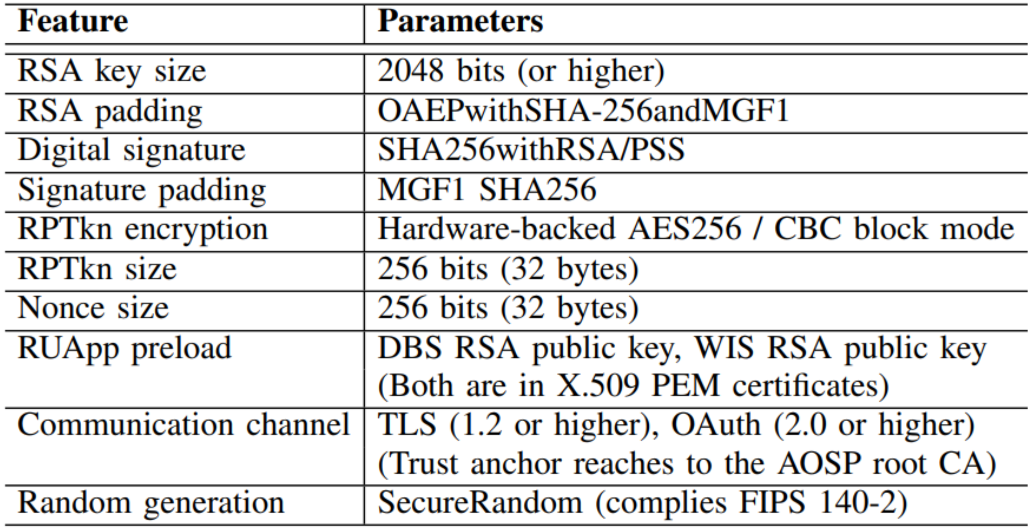


4. Implementation

Security Requirement

- Cryptographic specification observes NIST recommendations

- Application signing
  - Private key is not exposed from HSM
  - Achieves sandboxing, integrity, and permission system



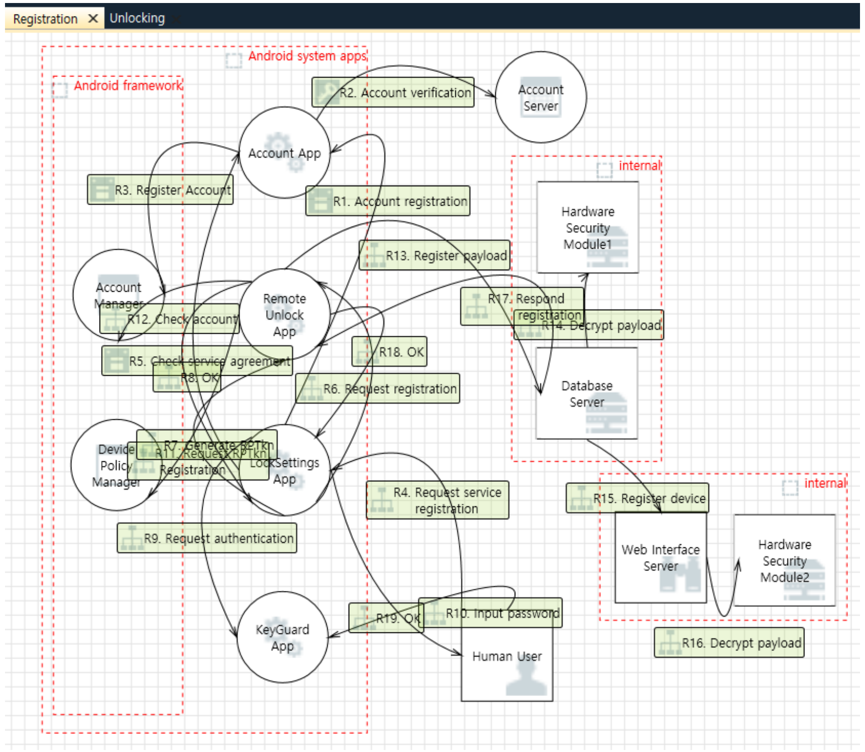
Feature	Parameters
RSA key size	2048 bits (or higher)
RSA padding	OAEPwithSHA-256andMGF1
Digital signature	SHA256withRSA/PSS
Signature padding	MGF1 SHA256
RPTkn encryption	Hardware-backed AES256 / CBC block mode
RPTkn size	256 bits (32 bytes)
Nonce size	256 bits (32 bytes)
RUApp preload	DBS RSA public key, WIS RSA public key (Both are in X.509 PEM certificates)
Communication channel	TLS (1.2 or higher), OAuth (2.0 or higher) (Trust anchor reaches to the AOSP root CA)
Random generation	SecureRandom (complies FIPS 140-2)

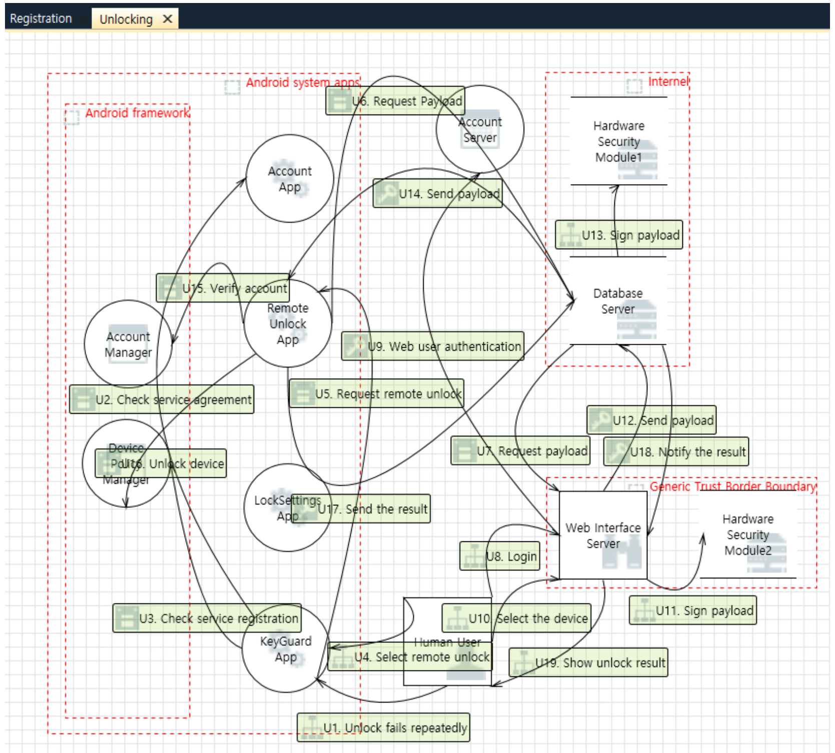
- Hide annotation (@hide) prevents 3rd party access from the API level 28
- Call stack monitoring prevents unrelated system app access to the RPTkn
- Custom permission requires explicit access: the accountability is achieved

5. Evaluation

Threat analysis

- We adopted STRIDE model by Microsoft Threat Modeling tool[2)]
  - Most mature model that helps identify relevant mitigating techniques

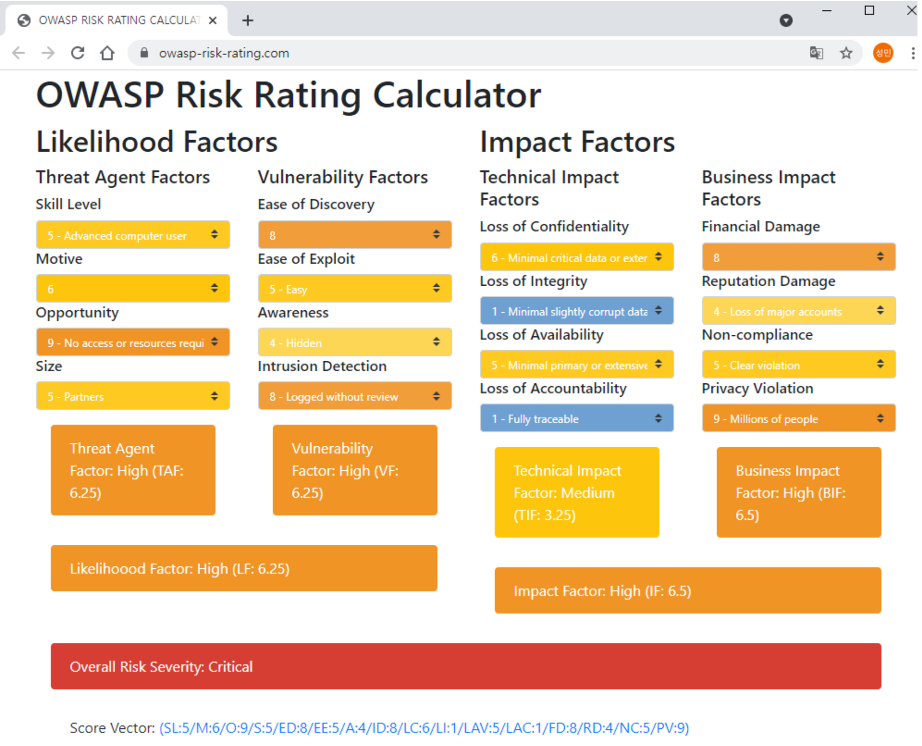




[2)] MSTM Data available: https://drive.google.com/file/d/1430prcH3Rx_Kd3TGIGZVyEhBTTQn3YVf

Risk assessment

- We adopted OWASP risk rating to assess threat severity
  - Found and evaluate 208 threats: HIGH(21), MEDIUM(46), LOW(141)



OWASP Risk Rating Calculator

Likelihood Factors

Threat Agent Factors

Skill Level: 5 - Advanced computer user

Motive: 6

Opportunity: 9 - No access or resources requi

Size: 5 - Partners

Threat Agent Factor: High (TAF: 6.25)

Vulnerability Factors

Ease of Discovery: 8

Ease of Exploit: 5 - Easy

Awareness: 4 - Hidden

Intrusion Detection: 8 - Logged without review

Vulnerability Factor: High (VF: 6.25)

Likelihood Factor: High (LF: 6.25)

Impact Factors

Technical Impact Factors

Loss of Confidentiality: 6 - Minimal critical data or exter

Loss of Integrity: 1 - Minimal slightly corrupt data

Loss of Availability: 5 - Minimal primary or extensive

Loss of Accountability: 1 - Fully traceable

Technical Impact Factor: Medium (TIF: 3.25)

Business Impact Factors

Financial Damage: 8

Reputation Damage: 4 - Loss of major accounts

Non-compliance: 5 - Clear violation

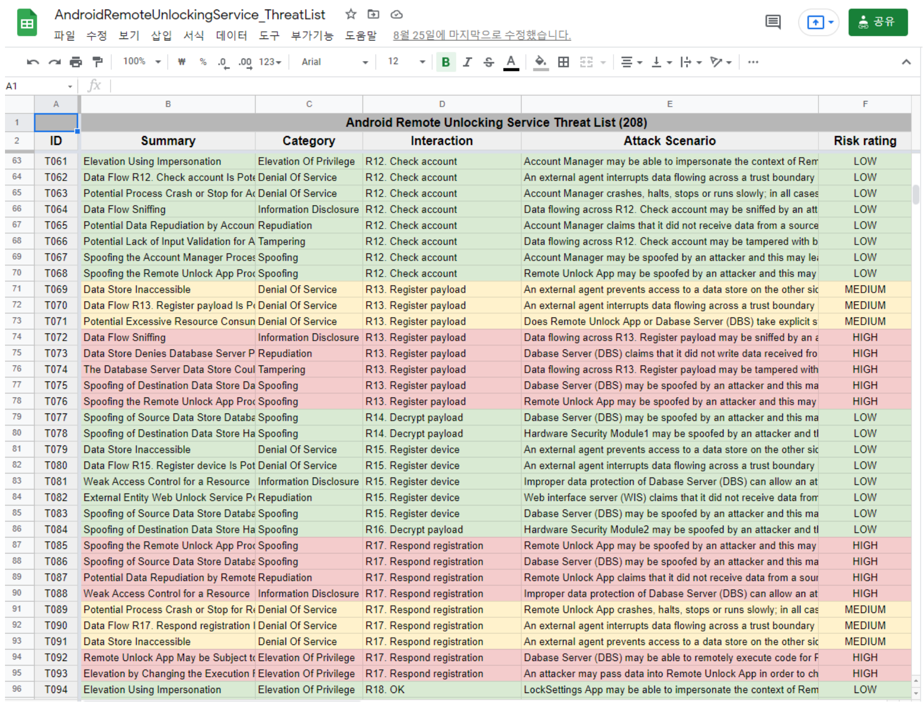
Privacy Violation: 9 - Millions of people

Business Impact Factor: High (BIF: 6.5)

Impact Factor: High (IF: 6.5)

Overall Risk Severity: Critical

Score Vector: (SL:5/M:6/O:9/S:5/ED:8/EE:5/A:4/ID:8/LC:6/LI:1/LAV:5/LAC:1/FD:8/RD:4/NC:5/PV:9)


AndroidRemoteUnlockingService_ThreatList
파일 수정 보기 삽입 서식 데이터 도구 부가기능 도움말
8월 25일에 마지막으로 수정했습니다.
공유
100%
123
Arial
12
A1
A
B
C
D
E
F

Android Remote Unlocking Service Threat List (208)

	ID	Summary	Category	Interaction	Attack Scenario	Risk rating
63	T061	Elevation Using Impersonation	Elevation Of Privilege	R12. Check account	Account Manager may be able to impersonate the context of Rem	LOW
64	T062	Data Flow R12. Check account Is Pote	Denial Of Service	R12. Check account	An external agent interrupts data flowing across a trust boundary	LOW
65	T063	Potential Process Crash or Stop for Ac	Denial Of Service	R12. Check account	Account Manager crashes, halts, stops or runs slowly; in all cases	LOW
66	T064	Data Flow Sniffing	Information Disclosure	R12. Check account	Data flowing across R12. Check account may be sniffed by an att	LOW
67	T065	Potential Data Repudiation by Accoun	Repudiation	R12. Check account	Account Manager claims that it did not receive data from a source	LOW
68	T066	Potential Lack of Input Validation for A	Tampering	R12. Check account	Data flowing across R12. Check account may be tampered with b	LOW
69	T067	Spoofing the Account Manager Proces	Spoofing	R12. Check account	Account Manager may be spoofed by an attacker and this may lea	LOW
70	T068	Spoofing the Remote Unlock App Proc	Spoofing	R12. Check account	Remote Unlock App may be spoofed by an attacker and this may	LOW
71	T069	Data Store Inaccessible	Denial Of Service	R13. Register payload	An external agent prevents access to a data store on the other sic	MEDIUM
72	T070	Data Flow R13. Register payload Is Po	Denial Of Service	R13. Register payload	An external agent interrupts data flowing across a trust boundary	MEDIUM
73	T071	Potential Excessive Resource Consun	Denial Of Service	R13. Register payload	Does Remote Unlock App or Dabase Server (DBS) take explicit s	MEDIUM
74	T072	Data Flow Sniffing	Information Disclosure	R13. Register payload	Data flowing across R13. Register payload may be sniffed by an a	HIGH
75	T073	Data Store Denies Database Server P	Repudiation	R13. Register payload	Dabase Server (DBS) claims that it did not write data received fro	HIGH
76	T074	The Database Server Data Store Coul	Tampering	R13. Register payload	Data flowing across R13. Register payload may be tampered with	HIGH
77	T075	Spoofing of Destination Data Store Da	Spoofing	R13. Register payload	Dabase Server (DBS) may be spoofed by an attacker and this ma	HIGH
78	T076	Spoofing the Remote Unlock App Proc	Spoofing	R13. Register payload	Remote Unlock App may be spoofed by an attacker and this may	HIGH
79	T077	Spoofing of Source Data Store Databa	Spoofing	R14. Decrypt payload	Dabase Server (DBS) may be spoofed by an attacker and this ma	LOW
80	T078	Spoofing of Destination Data Store Ha	Spoofing	R14. Decrypt payload	Hardware Security Module1 may be spoofed by an attacker and tl	LOW
81	T079	Data Store Inaccessible	Denial Of Service	R15. Register device	An external agent prevents access to a data store on the other sic	LOW
82	T080	Data Flow R15. Register device Is Pot	Denial Of Service	R15. Register device	An external agent interrupts data flowing across a trust boundary	LOW
83	T081	Weak Access Control for a Resource	Information Disclosure	R15. Register device	Improper data protection of Dabase Server (DBS) can allow an at	LOW
84	T082	External Entity Web Unlock Service Po	Repudiation	R15. Register device	Web interface server (WIS) claims that it did not receive data from	LOW
85	T083	Spoofing of Source Data Store Databa	Spoofing	R15. Register device	Dabase Server (DBS) may be spoofed by an attacker and this ma	LOW
86	T084	Spoofing of Destination Data Store Ha	Spoofing	R16. Decrypt payload	Hardware Security Module2 may be spoofed by an attacker and tl	LOW
87	T085	Spoofing the Remote Unlock App Proc	Spoofing	R17. Respond registration	Remote Unlock App may be spoofed by an attacker and this may	HIGH
88	T086	Spoofing of Source Data Store Databa	Spoofing	R17. Respond registration	Dabase Server (DBS) may be spoofed by an attacker and this ma	HIGH
89	T087	Potential Data Repudiation by Remote	Repudiation	R17. Respond registration	Remote Unlock App claims that it did not receive data from a sour	HIGH
90	T088	Weak Access Control for a Resource	Information Disclosure	R17. Respond registration	Improper data protection of Dabase Server (DBS) can allow an at	HIGH
91	T089	Potential Process Crash or Stop for Re	Denial Of Service	R17. Respond registration	Remote Unlock App crashes, halts, stops or runs slowly; in all cas	MEDIUM
92	T090	Data Flow R17. Respond registration I	Denial Of Service	R17. Respond registration	An external agent interrupts data flowing across a trust boundary	MEDIUM
93	T091	Data Store Inaccessible	Denial Of Service	R17. Respond registration	An external agent prevents access to a data store on the other sic	MEDIUM
94	T092	Remote Unlock App May be Subject to	Elevation Of Privilege	R17. Respond registration	Dabase Server (DBS) may be able to remotely execute code for F	HIGH
95	T093	Elevation by Changing the Execution F	Elevation Of Privilege	R17. Respond registration	An attacker may pass data into Remote Unlock App in order to ch	HIGH
96	T094	Elevation Using Impersonation	Elevation Of Privilege	R18. OK	LockSettings App may be able to impersonate the context of Rem	LOW

3) Full analysis result available: https://docs.google.com/spreadsheets/d/1wr7NPYgBpOH24OdeYgJB4hGTDWgJukJNfFuJ6XYb0Uw

Security Countermeasures

- High-level threats exist in the interaction between the device and the DBS
  - Proposed secure protocol defends all the High-level threats
  - Uses TLS, RSA, AES, and SHA256withRSA digital signatures

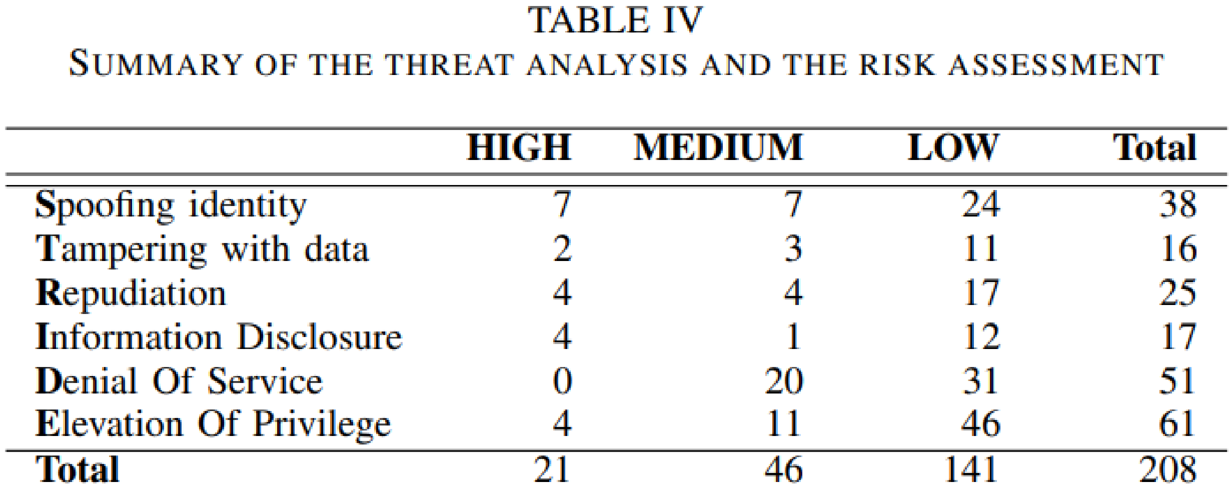


TABLE IV
SUMMARY OF THE THREAT ANALYSIS AND THE RISK ASSESSMENT

	HIGH	MEDIUM	LOW	Total
Spoofing identity	7	7	24	38
Tampering with data	2	3	11	16
Repudiation	4	4	17	25
Information Disclosure	4	1	12	17
Denial Of Service	0	20	31	51
Elevation Of Privilege	4	11	46	61
Total	21	46	141	208

- Verified that the almost Medium-level threats could be controlled
  - RUApp: Android custom permission, application signing
  - DPM: hide annotation, call-stack monitoring
  - WIS: OAuth 2.0, USAcnt locking (in the case of multiple login failures)

6. Conclusion

Conclusion

- Presented a new Android remote unlocking service
  - Proposed service can improve the user experiences but preserves Android h/w security

- Our design supports various security related features
  - two-factor authentication, distributed authority, trust-boundary minimization, key management, and compatibility

- Evaluated the security of the proposed remote unlocking service
  - Verified that our countermeasures defends all the identified high-level threats

- The service installed on commercial devices and launched in real world
  - After passing a manufacturer's quality verification and 3rd party penetration test

Thanks

Q&A