Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.31, NO.4, Aug. 2021 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2021.31.4.637

베이즈 네트워크를 이용한 탈중앙화 암호화폐 지갑의 정량적 위험성 평가

유 병 철,[1†] 김 승 주[2‡]
[1,2]고려대학교 정보보호대학원 (대학원생, 교수)

Quantitative Risk Assessment on a Decentralized Cryptocurrency Wallet with a Bayesian Network

Byeongcheol Yoo,[1†] Seungjoo Kim[2‡]
[1,2]ICSP(Institute of Cyber Security & Privacy), School of Cybersecurity, Korea University (Graduate student, Professor)

요 약

2009년 비트코인 블록체인이 처음 생성된 이후 암호화폐 사용자는 꾸준히 증가하고 있다. 하지만 이러한 사용자들의 암호화폐 지갑에 보관된 자산을 노리는 해킹 공격도 증가하고 있다. 따라서 우리는 시중에 나와 있는 암호화폐 지갑들이 안전하게 만들어졌는지를 점검하기 위해 각 지갑에 내재된 위험성을 평가한다. 우리는 위협 모델링을 통해 암호화폐 지갑에 내재된 위협을 식별하고 보안 요구사항을 도출한다. 그리고 도출된 보안 요구사항을 바탕으로 실제 지갑들의 보안성을 분석하고 공격트리와 베이즈 네트워크 등을 활용하여 각 지갑의 위험성을 정량적으로 측정한다. 위험성 평가 결과, 하드웨어 지갑보다 소프트웨어 지갑의 평균적인 위험성이 1.22배 높은 것으로 나타났다. 그리고 하드웨어 지갑 간 비교에서는 secure element를 내장한 Ledger Nano S 지갑보다 범용 MCU를 내장한 Trezor One 지갑의 위험성이 1.11배 높은 것으로 나타났다. 하지만 secure element를 사용하는 것은 암호화폐 지갑의 위험성을 낮추는 데에는 상대적으로 효과가 낮은 것으로 나타났다.

ABSTRACT

Since the creation of the first Bitcoin blockchain in 2009, the number of cryptocurrency users has steadily increased. However, the number of hacking attacks targeting assets stored in these users' cryptocurrency wallets is also increasing. Therefore, we evaluate the security of the wallets currently on the market to ensure that they are safe. We first conduct threat modeling to identify threats to cryptocurrency wallets and identify the security requirements. Second, based on the derived security requirements, we utilize attack trees and Bayesian network analysis to quantitatively measure the risks inherent in each wallet and compare them. According to the results, the average total risk in software wallets is 1.22 times greater than that in hardware wallets. In the comparison of different hardware wallets, we found that the total risk inherent to the Trezor One wallet, which has a general-purpose MCU, is 1.11 times greater than that of the Ledger Nano S wallet, which has a secure element. However, use of a secure element in a cryptocurrency wallet has been shown to be less effective at reducing risks.

Keywords: Risk Assessment, Threat Modeling, Cryptocurrency Wallet, Bayesian Network, Attack Tree

Received(04. 20. 2021), Modified(1st: 05. 21. 2021, 2nd: 06 .21. 2021), Accepted(06. 23. 2021)

† 주저자, architectophile@gmail.com
‡ 교신저자, skim71@korea.ac.kr(Corresponding author)

I. 서 론

2009년 비트코인[1] 블록체인이 최초로 생성된 이후 현재까지 암호화폐 사용자는 꾸준히 증가하고 있다. 현재 작성 시점을 기준으로 전 세계의 암호화폐 사용자는 5천만명 이상으로 추정된다. 특히 이더리움[2]에서 처음 소개한 스마트 컨트랙트[3] 덕분에 암호화폐의 발행과 관리가 쉬워지면서 암호화폐의 종류는 기하급수적으로 증가하였다.

하지만 암호화폐 사용자의 증가로 인해 암호화폐 자산에 대한 해킹 공격도 함께 증가하고 있다. 현재 작성 시점을 기준으로 전 세계의 암호화폐 해킹 피해 누적 금액은 12조 원 이상으로 추정된다. 특히 암호화폐 거래소에 대한 해킹 사고가 해마다 발생함에 따라 거래소가 아닌 개인용 암호화폐 지갑에서 직접 자산을 관리하려는 사용자들이 증가하고 있다. 하지만 그에 따라 사용자들의 개인용 암호화폐 지갑을 노리는 악성코드와 같은 다양한 해킹 공격도 함께 증가하고 있는 상황이다.

암호화폐 지갑 서비스는 기존의 전통적인 금융 서비스와 매우 다르다. 암호화폐 지갑 서비스와 전통적인 금융 서비스 사이의 결정적인 차이는 블록체인의 두 가지 특성인 익명성과 탈중앙화로 인해 발생한다. 블록체인 상의 암호화폐 계좌 주소는 사용자의 공개키를 이용해 생성된다. 그리고 그 공개키에 대응되는 개인키를 이용하여 사용자가 해당 계좌의 주인임을 입증한다. 이러한 방식에서는 블록체인 상의 계좌 정보를 사용자의 신원정보와 연결시키지 않기 때문에 매우 훌륭한 익명성을 제공한다. 그리고 블록체인 상의 거래 내역 장부는 탈중앙화된 방식으로 기록되고 관리된다. 따라서 블록체인 상의 동일한 거래 장부가 많은 수의 분산된 노드들에 의해 관리되기 때문에 블록체인 네트워크의 51% 이상을 장악하지 않는 이상 블록체인에 기록된 내용을 변경하는 것은 현실적으로 매우 어렵다. 따라서 블록체인에 기록된 거래 내역은 불변성과 무결성을 제공한다. 하지만 이러한 블록체인의 특성 때문에 암호화폐 지갑에 대한 공격 위협이 증가한다. 왜냐하면 해커가 사용자의 개인키를 훔친 뒤 사용자의 암호화폐 자산을 해커의 계좌에 전송하더라도 익명성 때문에 해당 계좌의 주인을 알기 어렵기 때문이다. 게다가 블록체인 장부의 불변성으로 인해 한 번 해커에게 전송된 암호화폐는 다시 돌려받을 수 없다. 그리고 만약 사용자가 자신의 암호화폐 개인키를 분실하였을 경우에는 해당 계좌의 자산을 다시 찾을 수 없다. 왜냐하면 개인키가 해당 계좌의 주인임을 입증할 수 있는 유일한 수단이기 때문이다. 따라서 이처럼 전통적인 금융 서비스와는 전혀 다른 특성을 갖고 있는 암호화폐 지갑의 보안성을 평가하기 위한 새롭고 체계적인 방법이 필요하다.

이러한 이유로 그동안 암호화폐 지갑의 보안성을 분석하기 위한 몇몇 연구들이 진행되었다 [4]-[8]. 하지만 이 연구들은 체계적인 방법을 이용한 보안성 평가보다는 일반적으로 알려진 암호화폐 지갑의 공격 벡터들을 정리하거나 특정 플랫폼 지갑이 갖고 있는 취약점에 대해 연구하는 경우가 대부분이었다. 그리고 [9], [10]은 위협 모델링 기법을 사용하여 좀 더 체계적으로 지갑에 존재하는 위협들을 분석하였으나 그 분석 방법과 과정에 대한 논리적인 근거나 설명이 부족했고 특정 플랫폼 지갑에 한정된 연구로 인해 다양한 형태의 지갑에는 적용하기 쉽지 않았다. 무엇보다도 지금까지는 암호화폐 지갑에 존재하는 위험성을 정량적으로 측정하고 평가하는 연구가 부족했다.

따라서 본 논문에서는 체계적으로 암호화폐 지갑의 위험성을 평가하는 방법을 제시한다. 우리는 위협 모델링 기법을 사용하여 암호화폐 지갑에 존재하는 위협들을 식별하고 보안 요구사항을 도출한다. 그리고 공격트리와 베이즈 네트워크, CVSS 메트릭을 비롯한 다양한 위험성 측정 요소들을 활용하여 실제 암호화폐 지갑들의 위험성을 정량적으로 측정한다. 그리고 일반적으로 안전하다고 알려진 하드웨어 지갑이 실제로 소프트웨어 지갑보다 안전한 지 확인하기 위해 각 지갑들의 위험성을 측정하여 비교한다. 또한 secure element를 내장하고 있는 하드웨어 지갑이 그렇지 않은 일반 하드웨어 지갑에 비해 얼마나 더 안전한 지 알아보기 위해 위험성을 서로 비교한다.

II. 배경 지식

2.1 암호화폐 지갑

암호화폐 지갑이란 사용자의 암호화폐를 관리하고 암호화폐의 전송 또는 수신 기능을 제공하는 소프트웨어 또는 하드웨어를 의미한다. 암호화폐 지갑은 사용자가 보유한 암호화폐 계좌에 대한 정보(주소, 잔고, 거래내역 등)를 제공하며, 사용자가 암호화폐를 전송할 때 새로운 거래(transaction) 데이터를 생성하여 이에 대한 서명을 생성한 뒤 블록체인 네트워크에 전파하는 역할을 한다.

Table 1. Taxonomy of cryptocurrency wallets based on key management, network connection, and platform.

Criteria	Key Management	Network Connection	Platform
Wallet	Decentralized Wallet	Hot Wallet (Software Wallet)	Mobile, PC, Web, Chrome Extension
		Cold Wallet (Hardware Wallet)	Embedded System, PC with no network connection
	Centralized Wallet	-	Crypto Exchange, Cloud

2.1.1 암호화폐 지갑 분류

우리는 암호화폐 지갑의 위험성을 평가하기에 앞서 각 지갑들의 특성에 따라 암호화폐 지갑을 분류하였다. Table 1은 암호화폐 지갑의 분류된 결과를 나타낸다. 암호화폐 지갑은 크게 탈중앙화된 지갑과 중앙화된 지갑 두 가지로 나눌 수 있다. 탈중앙화된 지갑은 암호화폐 키의 생명주기(생성, 변경, 사용, 폐기 등)를 사용자 장치에서 직접 관리하는 지갑을 말한다. 반면에 중앙화된 지갑은 암호화폐 키의 생명주기를 중앙화된 서버에서 관리하는 지갑을 말한다.

본 논문에서는 탈중앙화된 지갑에 대해서만 연구하였다. 그 이유는 탈중앙화된 지갑은 암호화폐 키를 사용자의 장치에서 직접 관리하기 때문에 블록체인 고유의 특성에 의해 기존의 전통적인 금융 서비스와는 전혀 다른 새로운 위협들이 발생하기 때문이다.

탈중앙화된 지갑은 다시 핫월렛과 콜드월렛 두 가지로 나눌 수 있다. 핫월렛은 암호화폐 지갑이 설치된 장치가 네트워크 인터페이스를 갖고 있어 언제든지 온라인 연결이 가능한 형태를 말한다. 예를 들어 모바일 기기(스마트폰) 또는 컴퓨터에 설치하여 사용되는 암호화폐 지갑들은 핫월렛으로 분류된다. 그리고 핫월렛은 일반적으로 소프트웨어 형태로 구현되기 때문에 소프트웨어 지갑이라고도 불린다. 반면에 콜드월렛은 네트워크 인터페이스가 없거나 온라인에 연결되지 않은 물리적으로 분리된 별도의 키 저장 공간을 갖고 있는 암호화폐 지갑을 말한다. 콜드월렛은 주로 별도의 하드웨어 장치로 구현되기 때문에 하드웨어 지갑이라고도 불린다.

2.2 관련 연구

최근 암호화폐에 대한 관심이 증가하면서 암호화폐 지갑의 위협 모델링을 통한 보안성 평가 연구들이 생겨나고 있다 [9] [10]. Zcash 암호화폐를 개발한 Electric Coin Company는 [9]에서 자신들이 만든 ECC 지갑에 대해 위협 모델을 작성하였다. 이들은 [11]에서 제안한 보안 불변사항 중심의 위협 모델링(Invariant-Centric Threat Modeling) 방법을 사용하여 ECC 지갑의 위협 모델을 작성하였다. 여기서 보안 불변사항이란 사용자가 안전하게 의존할 수 있다고 분석된 보안 속성을 말한다. 하지만 [9]는 ECC 지갑이 해당 작성된 보안 불변사항들을 보장한다는 사실을 어떤 방법을 통해 검증하였는지에 대한 구체적인 설명이 없었다. 게다가 공격 성공의 확률이나 위험성을 정량적으로 측정하는 것이 아니라 이처럼 보안 불변사항 리스트로만 표현하는 방식은 보안성 평가의 정확성에 문제가 발생한다.

보안 운영체제를 개발하는 Whonix 팀은 [10]에서 암호화폐 하드웨어 지갑의 위협 모델을 작성하였다. 여기서는 주로 하드웨어 지갑의 보안 디스플레이 기능의 중요성에 초점을 맞춰 외부의 호스트가 악성코드에 감염되었을 경우 어떤 위협들이 발생하는지 분석하였다. 이 위협 모델은 사용자가 매우 이해하기 쉬운 언어로 작성되어 있기 때문에 일반 하드웨어 지갑 사용자들이 참고하기에 매우 편리했다. 하지만 이 위협 모델은 발생 가능한 위협의 유형이 매우 제한적이고 보안 디스플레이 기능에 집중되어 있다 보니 펌웨어 변경이나 물리적 공격 등 좀 더 다양하고 구체적인 위협들에 대한 분석이 부족했다.

해킹 공격으로부터 안전하게 암호화폐 자산을 보관하기 위해 콜드월렛을 사용하는 사용자가 늘어나면서 콜드월렛에 대한 보안성 분석 연구도 진행되었다. M. Guri [8]은 망분리되어 보관되는 콜드월렛의 보안성을 분석하고, 콜드월렛 역시 은닉채널을 통해 개인키가 유출될 수 있음을 보여주었다. M. Guri는 서명된 거래를 전송할 때 감염된 USB 드라이버 등을 통해 콜드월렛 호스트가 악성코드에 감염될 수 있음을 설명하였으며, 은닉채널을 통해 데이터를 유출하는 악성코드인 브릿지웨어 [12]를 소개하였다.

D. Nedospasov 등 [13]은 Ledger와 Trezor의 하드웨어 지갑들의 전반적인 보안 취약점들을 소개하였다. [13]에서는 공급망 공격을 통해 Ledger Nano S 제품에 RF 트리거를 설치하여 원격으로 안테나를 이용해 사용자의 동의 없이 거래에 대한 서명을 승인하는 공격을 시연하였다. 그리고 Ledger Nano S 부트로더 취약점을 이용하여 프록시 MCU

에 임의의 펌웨어를 설치하는 공격을 보여주었다. 또한 Ledger Nano S에서 secure element의 펌웨어 검증을 우회하기 위한 펌웨어 압축 방법의 PoC를 소개하였다. 그리고 Trezor 지갑에 glitching 공격을 통해 STM32 MCU의 메모리 읽기 보호 메커니즘을 우회하여 JTAG 디버거를 이용해 RAM에 저장된 복구문구와 PIN을 탈취하는 공격을 소개하였다. 따라서 아무리 하드웨어 지갑이라도 공격자에게 물리적 접근과 충분한 시간만 제공된다면 해킹이 가능하다는 사실을 잘 보여주었다.

소프트웨어 지갑에 대한 보안성 분석 연구들도 진행되어 왔다. D. He 등 [7]은 안드로이드 운영체제에서 동작하는 암호화폐 지갑에 대해 취약점을 분석하고 이를 바탕으로 공격벡터를 설정하여 실제 두 개의 안드로이드 지갑들의 보안성을 분석하였다. 그리고 실험을 통해 두 개의 안드로이드 지갑에서 키보드 입력, 스크린 터치 입력, 스크린 출력 등의 정보를 수집하여 개인키 또는 복구문구와 같은 민감한 비밀정보를 얻어낼 수 있음을 보여주었다.

A. R. Sai 등 [4]는 모바일 지갑의 보안성과 프라이버시에 대해 분석하였다. 이 논문에서는 OWASP Mobile top 10을 기반으로 모바일 지갑의 위협을 도출하였다. 그리고 도출된 위협들에 대해서 소스코드 정적 분석과 네트워크 트래픽 검사, 그리고 자동 취약점 분석 툴을 이용해 시중의 모바일 지갑과 일반 금융 어플리케이션들을 분석하였다. 하지만 이 연구는 일반적인 OWASP 모바일 취약점을 기반으로 진행되었기 때문에 블록체인 특성으로 인해 발생하는 암호화폐 지갑의 고유한 위협에 대해서는 고려하지 않았다.

Er-Rajy, L 등 [5]는 비트코인 지갑에 대한 각종 위협에 대해 설명하였다. 예를 들면 웜이나 트로이 목마를 이용한 악성코드 공격을 이용한 개인키 탈취 공격 등에 대한 위협과 다양한 실제 사례들을 설명하였다. 또한 지갑 소프트웨어의 블록체인 네트워크와의 연결을 방해하는 서비스 거부 공격 등에 대한 위협도 소개하였다.

암호화폐 지갑의 보안성 분석뿐만 아니라 지갑의 보안성을 향상시키기 위한 방법들에 대한 연구들도 진행되어 왔다. [14], [15]는 ARM의 트러스트존(TrustZone) 기술을 활용한 방법을 제안하였다. G. Miraje 등 [14]는 오픈소스 지갑인 BitSafe 지갑을 기본 지갑으로 하고 여기에 추가적으로 트러스트존 기술을 적용하여 보안성을 높이는 방법을 소개하였다.

W. Dai 등 [15]는 트러스트존에 기반하여 Simplified Payment Verification(SPV) 지갑을 구현하는 방법을 제안하였다. 이 논문에서는 SPV 작업을 Secure Execution Environment (SEE)에서 동작하여 안전하게 블록체인 위에 기록된 거래를 검증하였다. 그리고 블록헤더를 SEE에서 암호화하여 Normal Execution Environment (NEE)에서 읽지 못하도록 하여 만약에 Rich OS가 악성코드에 감염되어도 SPV 작업은 안전하게 수행되도록 하였다. 키 생성, 서명 생성 등 모든 보안이 요구되는 기능들은 트러스트존의 SEE 안에서 동작되며 개인키는 secure storage에 저장된다. 게다가 보안 디스플레이 및 보안 터치스크린 드라이버를 구현하여 SEE 안에서 안전하게 사용자의 입력과 출력이 처리될 수 있도록 하였다.

Y. Liu 등 [16]은 암호화화폐 지갑의 안전한 키 생성 방법을 제안하였다. 이 방법은 랜덤한 시드값과 사용자의 비밀문구 문자열을 함께 SHA256 해시하여 개인키를 생성하는 방식이다. 이 방법은 사용자 장치에 저장된 시드값이 탈취되어도 사용자가 설정한 비밀문구를 알지 못한다면 개인키를 알 수 없기 때문에 키 관리의 안전성을 높여준다. 하지만 이 방법은 BIP 39 [17]의 비밀문구를 활용하면 동일하게 구현가능하다. 게다가 [16]은 생성한 시드값들을 잃어버릴 경우를 대비해 모든 시드값들을 백업해두어야 하는데, BIP 32 [18]의 마스터시드 한 개만 보관하면 모든 키들을 복구할 수 있는 방법에 비해 훨씬 불편하다는 단점이 있다.

III. 암호화폐 지갑 위험성 평가 방법

암호화폐 지갑 시스템위 위험성을 평가하는 방법은 크게 위협 모델링 단계와 위험성 측정 단계로 나뉜다. Fig.1은 전체 위험성 평가 과정을 나타낸다. 첫 번째 위협 모델링 단계에서는 암호화폐 지갑 시스템을 데이터 흐름도로 나타낸 후 STRIDE 분석을 통해 지갑 시스템에 내재된 위협을 도출한다. 그리고 공격트리를 작성하고 이를 바탕으로 보안요구사항 체크리스트를 작성한다. 두 번째로 위험성 측정 단계에서는 작성된 공격트리를 베이즈 네트워크로 변환한 후 시중 지갑들의 보안요구사항 체크리스트 분석 결과를 바탕으로 CVSS 및 다양한 위험성 측정 요소를 활용하여 각 지갑의 위험성을 측정한다.

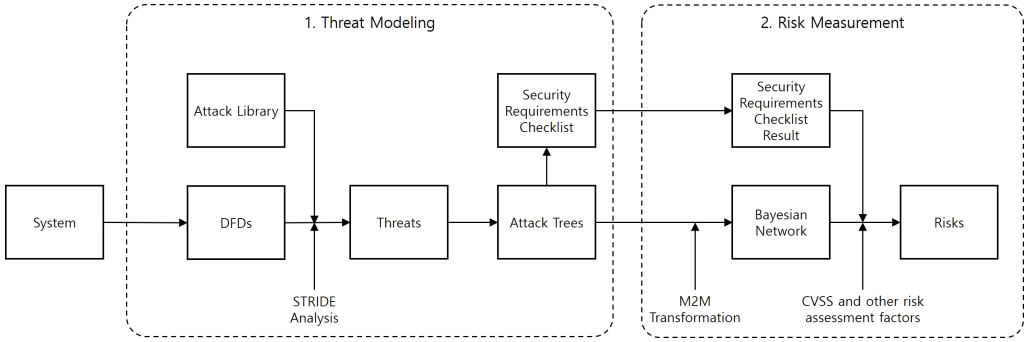

Fig. 1. The overall process of risk assessment.

3.1 위협 모델링

우리는 암호화폐 지갑 시스템에 내재된 위협을 도출하기 위해 위협 모델링을 사용한다. 위협 모델링은 대상 시스템을 모델링하여 체계적으로 보안 위협을 도출하는 기법이다. 우리는 위협 모델링 기법 중에서도 가장 잘 정립되었다고 알려진 Microsoft에서 개발한 STRIDE 기법을 사용한다. STRIDE 기법은 공격자의 관점에서 시스템의 각 컴포넌트에 발생 가능한 위협들을 위장(Spoofing), 변조(Tampering), 공격부인(Repudiation), 정보 노출(Information Disclosure), 서비스 거부(Denial of Service) 권한 상승(Escalation of Privilege)의 6가지 관점에서 분석하는 방법이다. 이를 통해 대상 시스템에 존재하는 위협들을 체계적이고 구체적으로 식별할 수 있다.

STRIDE 기법을 사용하기 위해서는 분석 대상 시스템을 프로세스 단위로 세분화하여 데이터 흐름도(DFD)를 작성하여 모델링하는 과정이 필요하다. 독립적인 기능을 수행할 수 있는 프로세스 단위로 시스템을 세분화함으로써 각 컴포넌트에 존재하는 위협을 구체적으로 분석할 수 있다.

시스템에 내재된 위협들을 도출한 후에는 해당 위협들을 활용하여 어떠한 공격 목표를 달성할 수 있는지 알아보기 위해 공격 트리를 작성한다. 이를 통해 각 공격에 사용되는 위협을 제거 또는 완화하기 위한 보안 요구사항을 도출할 수 있다.

3.1.1 데이트 흐름도(DFD) 작성

우리는 암호화폐 지갑의 구조를 분석하기 위해 데이터 흐름도를 작성하였다. Table 2는 DFD를 작성할 때 필요한 구성요소들을 나타낸다. 첫 번째 외부 개체(External Entity)는 대상 시스템 외부에 있는 사람 또는 시스템을 나타낸다. 일반적으로 어플리케이션을 사용하는 사용자 또는 외부 서버 등을 나타낸다. 그리고 저장 공간(Data Store)은 어플리케이션 안에서 데이터가 저장되는 공간을 나타낸다. 예를 들어 PC의 하드디스크 또는 스마트폰의 플래쉬 메모리 등이 해당된다. 그리고 프로세스(Process)는 어플리케이션 내에서 데이터를 처리하는 작업을 나타낸다. 입력 데이터를 처리하여 출력 데이터를 생성하는 프로세스 또는 실행 코드를 의미한다. 그리고 데이터 흐름(Data Flow)은 외부 개체, 저장 공간, 프로세스 사이에서 주고받는 데이터의 흐름을 나타낸다. 입력 데이터 또는 출력 데이터 등에 해당한다. 마지막으로 신뢰 경계(Trust Boundary)는 구성요소 간의 신뢰 수준이 달라지는 경계를 나타낸다. 신뢰 경계를 가로지르는 데이터 흐름은 기본적으로 신뢰할 수 없다고 간주한다. 예를 들어 어떤 시스템이 네트워크를 통해 외부 서버와 통신할 경우 서로 간에 주고받는 데이터는 신뢰할 수 없기 때문에 신뢰 경계를 통해 나타낸다.

Table 2. Elements of data flow diagram.

Element	Symbol	Description
External Entity	External Entity	Any entity(people or system) outside of the application
Data Store	Data Store	A location where data are stored
Process	Process	A task that handles data within the application
Data Flow	→	A path that data take between external entities, processes and data stores
Trust Boundary	⌒	The change of trust levels as data flow through the system

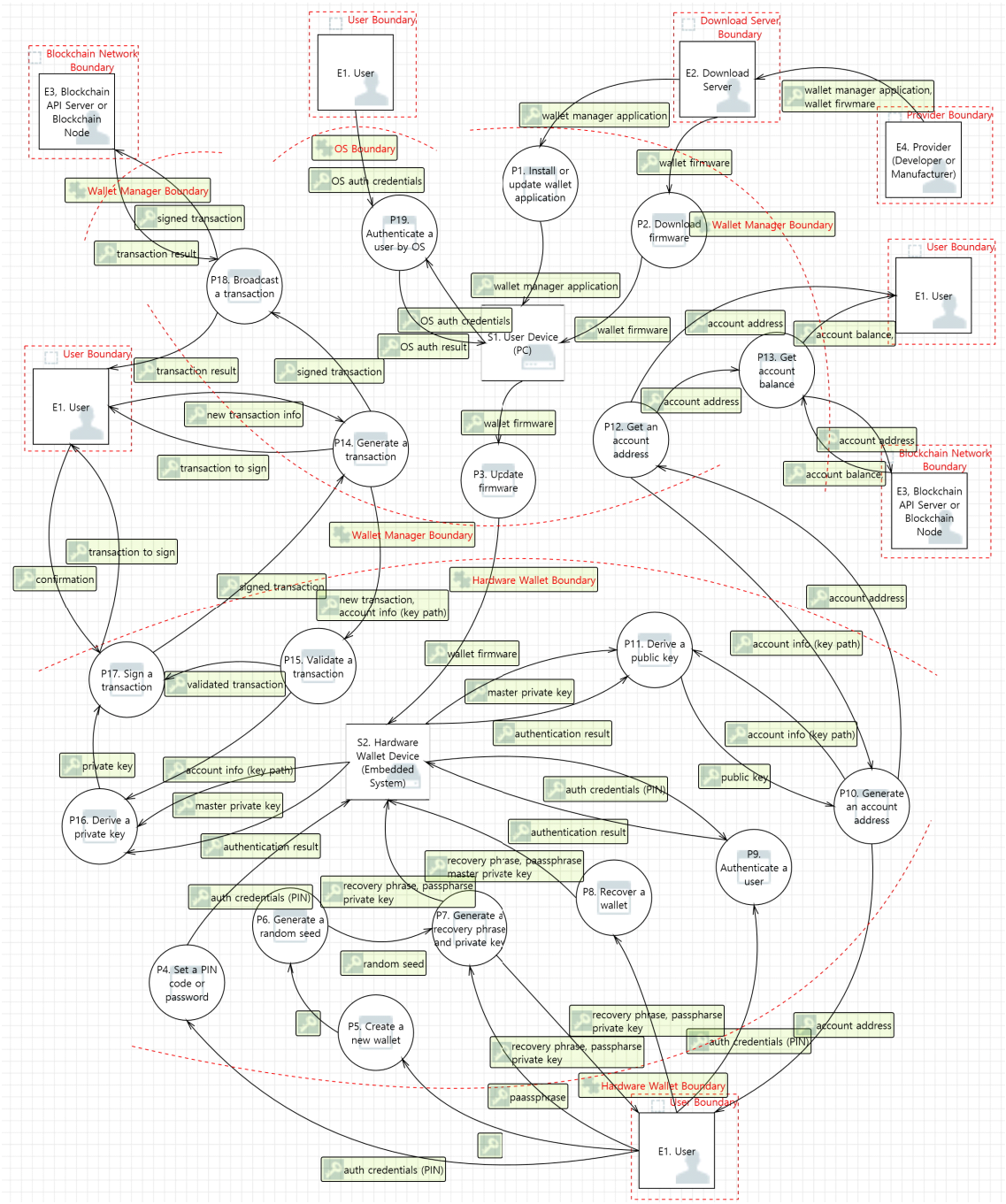

Fig. 2. Level 2 data flow diagram of a cold wallet.

데이터 흐름도는 추상화 정도에 따라서 다르게 표현된다. 레벨 0은 대상 시스템을 가장 추상화된 형태로 나타낸 것이며 레벨이 높을수록 더욱 구체적으로 시스템을 모델링한다. 일반적으로 레벨 2까지 표현하며 각 프로세스가 독립적인 기능을 하는 매우 구체적인 형태로 표현된다. 우리는 Microsoft에서 제공하는 Threat Modeling Tool을 사용하여 DFD를 작성하였다.

Table 3. Attack library for a cryptocurrency wallet.

Category	Attack	Num	Title	Author	Type
Bypassing Authentication	Brute-force attack on a PIN or password	1	Brute-force and dictionary attack on hashed real-world passwords	L. Bošnjak et al.	Conference
	Buffer overflow (code reuse)	2	Jump-oriented programming: a new class of code-reuse attack	Bletsch, Tyler, et al.	Conference
(Omitted)					
Malware	Clipboard hijacker	14	Simple Clipboard Malware Attack Detection and Analysis from the User-Machine Interaction View	Wieczorka, Michał	Conference
	Keylogger (keyboard, mouse, screen touch input logger)	15	A framework for detection and prevention of novel keylogger spyware attacks	M. Wazid et al.	Conference
(Omitted)					
Physical Attack	Fault injection (glitching)	33	wallet.fail	Dmitry Nedospasov et al.	Conference
	Microscopy	34	Reverse engineering flash EEPROM memories using scanning electron microscopy	Courbon et al.	Conference
	Probing	35	A layout-driven framework to assess vulnerability of ICs to microprobing attacks	Q. Shi, N et al.	Conference
	Side-channel attack	36	Extracting the Private Key from a TREZOR	Jochen Hoenicke	Public
Privilege Escalation	Android root toolkit	37	Android rooting: Methods, detection, and evasion	Sun, San-Tsai et al.	Conference
	Buffer overflow (code injection)	38	Code injection attacks on harvard-architecture devices	Francillon et al.	Conference
	Row Hammer attack	39	Android data storage security: A review	Altuwaijri et al.	Journal
			The Rowhammer Attack Injection Methodology	K. S. Yim	Conference

Fig.2는 콜드월렛의 DFD 레벨 2를 나타낸다. DFD를 통해 사용자 인증정보, 복구문구, 개인키 등과 같은 중요 데이터가 어디에서 어디로 흐르는지 상세하게 파악할 수 있기 때문에 지갑에 존재하는 위협들을 식별할 수 있다. 예를 들어 사용자가 최초 지갑을 생성하는 프로세스에서 출력되는 복구문구 데이터를 패시브 공격을 통해 가로챌 수 있다면 공격자는 키 유도 함수를 이용해 사용자의 모든 암호화폐 자산의 개인키를 탈취할 수 있다.

3.1.2 STRIDE 분석

STRIDE 기법은 각 컴포넌트에 발생 가능한 위협들을 위장(S), 변조(T), 공격부인(R), 정보 노출(I), 서비스 거부(D), 권한 상승(E)의 6가지 유형으로 분류하여 분석하는 위협 모델링 방법이다. STRIDE 분석 절차는 다음과 같다.

첫 번째는 암호화폐 지갑 시스템에 위협이 될 수 있는 알려진 공격 기법들 또는 취약점을 수집하여 공격 라이브러리를 작성한다. Table 3은 저널, 컨퍼런스, 책, CVE 및 CAPEC 등을 참조하여 작성한 공격 라이브러리를 나타낸다.

두 번째는 Table 3의 공격 라이브러리를 바탕으로 3.1.1장에서 작성된 DFD의 모든 컴포넌트를 대상으로 STRIDE 6가지 유형에 대해 발생 가능한 위협을 식별한다. 예를 들어 Fig.2의 E1.User에 대해서는 공격자가 사용자로 위장하여 지갑을 사용할 수 있다. 따라서 공격 라이브러리의 Bypassing Authentication의 공격 기법들을 활용하여 위장 공격이 가능하므로 Table 4의 C1 위협이 도출된다. Table 4는 콜드월렛의 DFD 레벨 2를 바탕으로 작성된 각 컴포넌트의 STRIDE 분석 결과를 나타낸다. 분석 결과, 콜드월렛에서는 103개의 위협이 도출되었고, 핫월렛에 대해서 총 112개의 위협이 도출되었다.

3.1.3 공격트리 작성

공격트리는 B. Schneier [19]가 제안한 것으로서 대상 시스템을 공격하기 위한 다양한 공격 경로를 체계적으로 도출할 수 있는 위협 모델링 방법이다. 우리는 3.1.2장에서 STRIDE 기법을 이용하여 도출한 위협들을 바탕으로 공격트리의 각 공격목표를 달성하기 위한 공격 시나리오를 작성하였다. 공격트리에서 루트 노드는 최종 목표(goal)를 나타내며, 해당 목표를 달성하기 위한 하위 목표(subgoal)들

Table 4. STIRDE analysis result of a cold wallet.

Component	Name	STRIDE	Num	Threat	Attack Library
Entity	E1. User	S	C1	Impersonate a user by bypassing wallet user authentication.	1, 2, 3, 4, 5, 6, 33
		S	C2	Impersonate a user by bypassing OS authentication.	1, 2, 3, 4, 5, 6, 33
(Omitted)					
Data Store	S2. Hardware Wallet	T	C24	Modify firmware, auth credentials, a recovery phrase, passphrase or private key by connecting a debugger (JTAG, SWD)	32
		I	C25	Obtain auth credentials, a recovery phrase, passphrase or private key using physical attacks (e.g., fault injection, probing, microscoping or cold boot attack).	31, 33, 34, 35, 36
		I	C26	Obtain auth credentials, a recovery phrase, passphrase or private key by connecting a debugger (JTAG, SWD)	32
(Omitted)					
Process	P17. Sign a transaction	T	C87	Modify a transaction by installing a modified firmware using social engineering or supply chain attack.	21, 22, 23, 24, 25, 26, 27
		I	C88	Obtain a private key using side channel attacks.	36
		I	C89	Compute a private key using an ECDSA nonce exploitation.	9, 10
		I	C90	Obtain transaction information using shoulder-surfing attack.	6
	P18. Broadcast a transaction	S	C91	Impersonate a normal blockchain node or API server using MITM attacks.	28, 29, 30
		T	C92	Modify a transaction using MITM attacks.	28, 29, 30
		I	C93	Obtain transaction information using MITM attacks.	28, 29, 30
		I	C94	Obtain transaction information by installing a screen recorder.	18
		D	C95	Prevent the wallet from broadcasting a transaction using MITM attacks.	28, 29, 30
		D	C96	Prevent the wallet from broadcasting a transaction by executing DoS attacks on the blockchain node or API server.	11, 12, 13, 17
	P19. Authenticat e a user by OS	S	C97	Bypass OS authentication using a brute-force attack	1
		S	C98	Bypass OS authentication using a buffer overflow (code reuse) attack.	2
		S	C99	Bypass OS authentication using evil maid attack.	3
		S	C100	Bypass OS authentication using fake biometrics.	4
		S	C101	Bypass OS authentication by accessing the wallet when it is unlocked.	5
		S	C102	Bypass OS authentication using shoulder-surfing attack.	6
		S	C103	Bypass OS authentication using physical attacks (e.g., fault injection(glitching)).	33

을 자식 노드로 갖는다. 그리고 중간에 여러 개의 브랜치 노드(branch node)를 거쳐 마지막 리프 노드는 실질적인 공격 위협(threat)을 나타낸다.

공격트리의 장점은 하나의 공격 목표를 달성하기 위해 필요한 다양한 구체적인 공격벡터들을 논리적이고 체계적으로 도출할 수 있다는 점이다. 우리는 암호화폐 지갑에 대한 공격자의 최종 공격 목표를 크게 세 가지로 나누었다. 암호화폐 자산 탈취, 서비스 거부 공격, 그리고 프라이버시 침해이다.

3.1.3.1 암호화폐 탈취 목표 (G1)

Table 5의 G1은 암호화폐 탈취 목표의 공격트리를 나타낸다. 첫 번째 하위 목표 S1은 사용자의 개인키를 얻어내는 것이다. 만약 해당 지갑이 계층 결정적 지갑 [17]일 경우에는 복구문구를 탈취하는 것은 개인키를 탈취하는 것과 동일하다. 이를 위해서는 사용자 장치에 악성코드를 설치하여 지갑의 입출력 데이터를 모니터링하거나 또는 클립보드에 복사된 데이터를 스니핑하는 공격 또는 장치에 저장된 키를 읽어오는 공격 등이 가능하다. 이러한 대부분의 공격들은 사용자 장치에 악성코드를 설치함으로써 수행이 가능하다. 만약 공격자가 물리적으로 사용자 장치에 접근할 수 있다면, 사용자 인증을 우회하거나 물리적 공격을 통해 키를 탈취할 수 있다. 그리고 공급망(supply chain) 공격을 통해 공격자에게 미리 알려진 키를 생성하도록 지갑을 변조하는 공격이나 또는 지갑에서 취약한 난수 발생기를 사용할 경우에는 서명의 넌스(nonce)값을 통해 개인키를 알아내는 공격 등이 가능하다 [18].

두 번째 하위 목표인 S2는 사용자 지갑 장치에서 공격자에게 암호화폐를 전송하는 것이다. 이를 위해 사용자를 속여서 잘못된 거래를 승인하도록 만들거나 또는 공격자가 직접 장치에 물리적으로 접근하여 사용자 인증을 우회하는 방법 등이 있다. 사용자를 속이기 위한 가장 간단한 방법은 사용자가 거래를 생성하기 위해 상대방의 주소를 클립보드에 복사했을 때 이를 공격자의 주소로 바꿔치기 하는 방법이다. 대부

Table 5. Attack trees for a cryptocurrency wallet.

Description								STRIDE Analysis	Node
Stealing Cryptocurrency									G1
1	Obtain a private key								S1
OR	1.1	Eavesdrop input data							B1
	OR	1.1.1	Keylogger malware						B2
		AND	1.1.1.1	Install a malware (keylogger, screen touch input logger)					B3
			OR	1.1.1.1.1	Social engineering (malicious files, malvertising, phishing, drive-by download attack)			H18	T1
				1.1.1.1.2	Rogue AP			H19	T2
				1.1.1.1.3	Supply chain attack			H20	T3
				1.1.1.1.4	Removable media (USB drive)			H21	T4
			1.1.1.2	Execute keylogging attack				H51	T5
	1.2	Eavesdrop output data							B4
	OR	1.2.1	Screen capture malware						B5
		AND	1.2.1.1	Install a malware (screen recorder)					B6
			OR	1.2.1.1.1	Social engineering (malicious files, malvertising, phishing, drive-by download attack)			H18	T6
				1.2.1.1.2	Rogue AP			H19	T7
				1.2.1.1.3	Supply chain attack			H20	T8
				1.2.1.1.4	Removable media (USB drive)			H21	T9
			1.2.1.2	Execute screen capture attack				H43, H49, H97	T10
	1.3	Observe output data directly on the screen							B7
(Omitted)									
Denial of Service									G2
4	Prevent a user from using a private key								S4
OR	4.1	Delete a private key							B68
	OR	4.1.1	Open the wallet and delete a private key						B69
		AND	4.1.1.1	Bypass OS authentication					B70
			OR	4.1.1.1.1	Brute-force attack (guessing, dictionary attack)			H106	T188
				4.1.1.1.2	Buffer overflow (code reuse)			H107	T189
				4.1.1.1.3	Evil maid attack			H108	T190
				4.1.1.1.4	Fake biometrics			H109	T191
				4.1.1.1.5	Physical access when the host is open			H110	T192
				4.1.1.1.6	Shoulder-surfing attack			H111	T193
				4.1.1.1.7	Physical attack (fault injection(glitching))			H112	T194
			4.1.1.2	Bypass wallet user authentication					B71
			OR	4.1.1.2.1	Brute-force attack (guessing, dictionary attack)			H53, C50	T195
				4.1.1.2.2	Buffer overflow (code reuse)			H54, C51	T196
				4.1.1.2.3	Evil maid attack			H55, C52	T197
				4.1.1.2.4	Fake biometrics			H56	T198
				4.1.1.2.5	Physical access when the wallet is open			H57, C53	T199
				4.1.1.2.6	Shoulder-surfing attack			H58, C54	T200
				4.1.1.2.7	Physical attack (fault injection(glitching))			H59, C55	T201
				4.1.1.2.8	Obtain auth credentials using a malware (keylogger, screen recorder)			H35, H36, H37, H60, H61	T202
(Omitted)									
Privacy Breach									G3
8	Obtain a user's personally identifiable information								S8
OR	8.2	Obtain personal information when a user uses the wallet							B115
	OR	8.2.4	Eavesdrop network traffic						B125
		OR	8.2.4.1	Network packet sniffer					B126
			AND	8.2.4.1.1	Install a malware (network packet sniffer)				B127
				OR	8.2.4.1.1.1	Social engineering (malicious files, malvertising, phishing, drive-by download attack)		H18	T328
					8.2.4.1.1.2	Rogue AP		H19	T329
					8.2.4.1.1.3	Supply chain attack		H20	T330
					8.2.4.1.1.4	Removable media (USB drive)		H21	T331
				8.2.4.1.2	Execute network packet sniffing attack				H102	T332
			8.2.4.2	Man-in-the-middle attack					B128
			OR	8.2.4.2.1	ARP spoofing			H99, H101	T333
				8.2.4.2.2	DNS spoofing and poisoning			H99, H101	T334
				8.2.4.2.3	IP address spoofing			H99, H101	T335

분 암호화폐 주소 형식은 사용자가 읽기 어렵기 때문에 이처럼 주소를 바꿔치기 하는 공격은 매우 효율적인 방법이다.

세 번째 하위 목표 S3은 사용자에게 전송될 암호화폐를 공격자가 가로채는 것이다. 가장 쉬운 방법은 두 번째 하위 목표에서 사용한 공격과 비슷하게 클립보드에 저장된 사용자의 주소를 공격자의 주소로 바꿔치기 하는 것이다. 이것은 사용자가 제 3자로부터 암호화폐를 전송받기 위해 자신의 주소를 건네줄 때 클립보드에 저장된 주소를 공격자의 주소로 바꿔치기 하는 방법으로서 매우 간단하고 효율적인 방법이다.

3.1.3.2 서비스 거부 공격 목표 (G2)

Table 5의 G2는 서비스 거부 공격 목표의 공격트리를 나타낸다. 첫 번째 하위 목표 S4는 사용자의 개인키에 대한 접근을 막는 것이다. 이를 위해서는 개인키를 삭제하는 공격이 가능하다. 사용자 장치에 악성코드를 설치하거나 루트 또는 관리자 권한을 획득하여 저장된 키를 삭제할 수 있다. 또는 물리적으로 접근이 가능한 경우 사용자 인증을 우회하거나 디스크 포맷 또는 팩토리 리셋 등으로 키를 삭제할 수 있다. 이 밖에도 고의적으로 사용자 인증을 연속으로 실패해 지갑에서 키를 삭제하도록 유도하거나 랜섬웨어를 이용해 키를 암호화하는 공격 등이 있다.

두 번째 하위 목표인 S5는 사용자의 어플리케이션 접근을 막는 것으로서 어플리케이션을 삭제하거나 암호화하는 방법 등이 가능하다. 이것은 S4에서의 공격 방법들과 유사하다.

세 번째 하위 목표인 S6는 지갑의 블록체인 네트워크 접속을 막는 것이다. 이를 위해서는 중간자 공격 등을 통해 정상적인 블록체인 네트워크와의 연결을 방해하는 방법이 가능하다. 또는 지갑이 특정 블록체인 API 서버를 통해 블록체인 데이터를 주고받을 경우에는 해당 서버에 서비스 거부 공격을 수행하여 지갑과의 연결을 방해할 수 있다.

3.1.3.3 프라이버시 침해 목표 (G3)

Table 5의 G3는 프라이버시 침해 목표의 공격트리를 나타낸다. 첫 번째 하위 목표 S7은 사용자 암호화폐 계좌의 정보를 얻어내는 것이다. 이를 위해서는 악성코드를 설치하여 지갑의 입출력 데이터를 모니터링 하거나 또는 네트워크 패킷을 스니핑하여 계좌 정보를 얻어낼 수 있다. 대부분 블록체인의 특성상 사용자의 계좌 정보만 알아내면 잔액뿐만 아니라 해당 계좌와 관련된 다른 계좌와의 거래 정보도 알아낼 수 있기 때문에 사용자의 계좌 주소만 노출되어도 상당히 많은 정보가 노출된다는 특징이 있다.

두 번째 하위 목표 S8은 사용자의 식별가능한 개인정보를 탈취하는 것이다. 이를 위해서는 S7에서와 비슷한 공격 방법이 가능하다. 하지만 일반적으로는 탈중앙화된 암호화폐 지갑들은 사용자의 개인정보를 입력받거나 저장하지 않기 때문에 개인정보 유출에 대한 위협이 많지 않다.

3.2 위험성 측정 방법

우리는 암호화폐 지갑들의 위험성을 측정하기 위해서 작성된 공격트리를 베이즈 네트워크(Bayesian Network)로 변환하고 CVSS Exploitability 메트릭 및 다양한 위험측정 기준 요소들을 활용한다.

우리는 지갑의 위험성을 측정하기 위한 방법을 마련하기 위해서 전 세계의 표준기관 또는 국가기관 등에서 만든 다양한 사이버보안 위협에 대한 위험관리 또는 위험평가 표준들을 참조하였다. 그 중에서 위험관리 또는 위험성 평가 관련된 논문들에서 공통적으로 언급되는 OCTAVE Allegro, NIST RMF, ISO/IEC 27005, CVSS, SP 800-30, FAIR, CRAMM, EBIOS, ISRAM, CORAS, COBIT 5, MEHARI 총 12개의 표준들을 참조하였다.

이 중에서 NIST RMF(SP 800-37), ISO/IEC 27005와 같은 상위레벨 표준들은 위험성 측정을 위한 하위레벨의 구체적인 방법은 명시하지 않는다. 반면에 SP800-30, CVSS, FAIR, EBIOS 등은 위험성을 측정하기 위한 기준 요소를 제공한다. 예를 들어 CVSS는 특정 위협의 심각도를 측정하기 위해 위협의 악용 가능성을 나타내는 Exploitability 메트릭과 공격성공의 결과로 나타나는 Impact 메트릭을 이용하여 심각도의 점수를 계산한다. FAIR는 공격 시나리오를 기반으로 위협의 발생 가능성을 나타내는 Loss Event Frequency 항목과 그로 인해 발생하는 피해 규모를 나타내는 Loss Magnitude 항목을 계산하여 전체 위험성을 정량적으로 측정한다. 그리고 EBIOS에서는 공격자의 Motivation, Resources, Activity를 고려하여 공격의 발생 가능성을 나타내는 likelihood를 구한 다음 공격의 심각도를 나타내는 severity와 함께 매트릭스로 계산

하여 위험성을 측정한다. 하지만 모든 이러한 방법들은 위험성을 측정할 때 각 개별적인 위협들이 서로 결합되어 공격에 사용되는 구조적인 컨텍스트(context)를 반영하지 못한다는 문제점이 있다. 하나의 동일한 위협이 존재하더라도 대상 시스템의 구조에 따라서 그리고 다른 어떤 위협과 결합할 수 있느냐에 따라서 측정되는 위험이 달라야 한다. 이를 위해서 CVSS에서는 Scope 메트릭을 사용하여 해당 위협이 다른 컴포넌트에도 영향을 미칠 때 점수를 다르게 계산하도록 하였지만, 실제로 어떤 컴포넌트와 어떻게 연결되었는지 구조적인 컨텍스트를 입력하는 것은 아니기 때문에 이 방법은 부정확하다.

우리는 이전 장에서 공격트리를 작성하여 서로 다른 위협들이 연결되는 구조를 나타내었다. 따라서 작성된 공격트리를 입력으로 하여 각 위협들의 결합 구조를 반영하는 위험성 측정 방법이 필요하다.

3.2.1 공격트리의 베이즈 네트워크 변환

베이즈 네트워크는 확률 그래프 모델로서 유향 비순환 그래프(Directed Acyclic Graph, DAG) 구조를 갖는다. 베이즈 네트워크의 각 노드는 확률변수를 나타내며, 서로를 잇는 방향성을 가진 엣지는 확률변수 간의 조건부 의존성(conditional dependency)을 나타낸다. 그리고 베이즈 네트워크를 이용하면 주어진 어떤 증거를 바탕으로 확률적인 베이즈 추론을 할 수 있다. 예를 들면 어떤 증상들이 나타났을 때 특정 질병을 가질 확률을 계산할 수 있다. 따라서 베이즈 네트워크는 인공지능, 질병진단, 문서분류 등 다양한 분야에서 널리 사용되고 있으며, 특히 사이버보안 분야에서도 위협 탐지, 스팸 필터링 등에 사용되고 있다. 그리고 [22], [23]는 결함트리를 베이즈 네트워크로 변환하여 시스템의 오류가 발생할 확률을 계산하였으며, [24]는 서로 다른 공격트리를 하나의 베이즈 네트워크로 변환하여 공격의 발생 가능성을 계산하였다. 따라서 우리는 암호화폐 지갑에서 서로 다른 위협들이 결합되었을 때 위험성을 측정하기 위해서 [22-24]에서 사용된 Model-to-Model (M2M) 변환 방법을 통해 우리가 작성한 공격트리를 베이즈 네트워크로 변환하여 공격의 발생 확률을 계산한다. 그리고 계산된 공격 발생 확률을 바탕으로 일반적으로 알려진 위험성 계산 공식 (1)을 이용하여 각 공격에 대한 위험성을 측정한다.

$$Risk = Probability \times Impact \quad (1)$$

Fig.3은 공격트리의 AND, OR 연산에 대한 베이즈 네트워크 변환 방법을 보여준다. 그림을 보면 변환 후에 부모 자식 관계가 뒤바뀌는 것을 알 수 있다. 그리고 베이즈 네트워크의 각 노드는 베르누이 확률 변수 x = {1, 0}를 의미한다. 따라서 각 노드는 0 또는 1의 상태를 갖으며, 0은 해당 위협이 발생하지 않았음을 나타내고, 1은 해당 위협이 발생하였음을 나타낸다. 그리고 조건부 확률 테이블(CPT)을 살펴보면 OR 연산일 때는 한 개 이상의 부모 노드의 상태가 1이면 자식 노드의 상태가 1이 되고, AND 연산일 때는 모든 부모 노드의 상태가 1이 되어야 자식 노드의 상태가 1이 되는 것을 볼 수 있다.

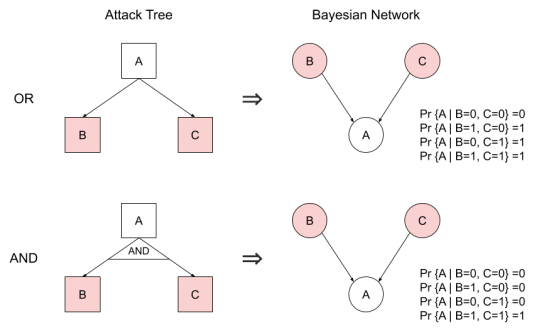

Fig. 3. Attack tree to Bayesian network translation using the M2M transformation method. Conditional probability tables(CPT) are located next to BN models.

Fig.4의 왼쪽에는 3.1.3장에서 작성한 G1, G2, G3 세 가지 목표에 대한 각 공격트리가 간략하게 나타나 있다. 그리고 오른쪽에는 해당 공격트리들을 M2M 방법을 통해 변환한 베이즈 네트워크가 나타나 있다. 이 때 주목할 부분은 베이즈 네트워크는 동일한 노드끼리 서로 결합할 수 있다는 점이다. 베이즈 네트워크는 동일한 부모 노드를 서로 다른 자식 노드끼리 공유할 수 있기 때문에 동일한 위협을 나타내는 노드들을 하나의 노드로 결합하여 표현할 수 있다. Fig.4에서 점선으로 표시된 노드들은 다른 노드와 결합할 수 있는 노드를 나타낸다. 예를 들어 G1 공격트리의 B1 노드는 입력 데이터를 도청하여 사용자의 개인키를 탈취하는데 이용될 수 있다. 그리고 G3의 B96 노드는 입력 데이터를 도청하여 사용자

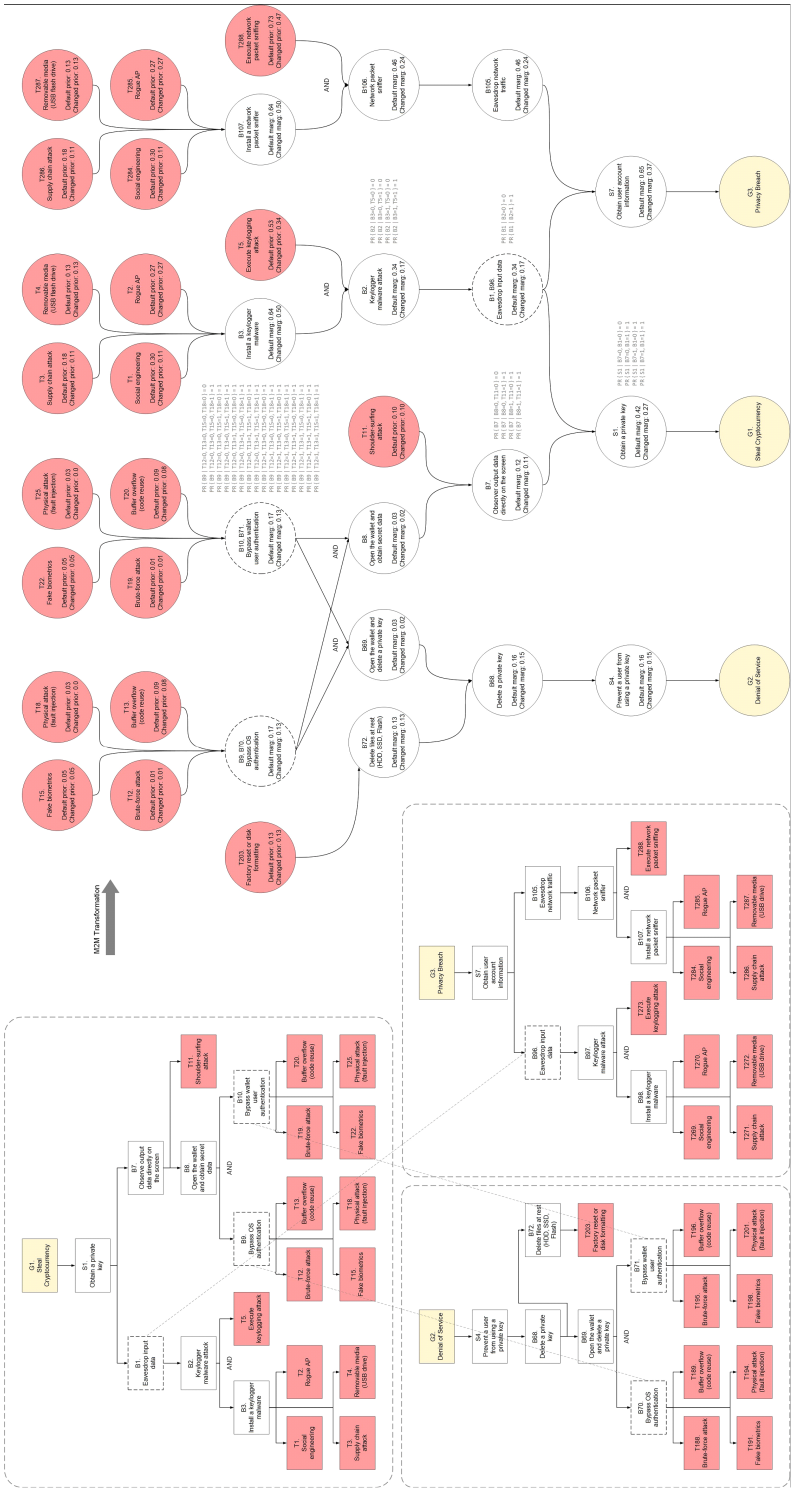

Fig. 4. Translation of attack trees (G1, G2, and G3) to a Bayesian network using the M2M transform method.

Table 6. Risk assessment factors from the standards.

Method	CVSS Exploitability	SP 800-30	OCTAVE Allegro	FAIR	EBIOS	CRAMM	MEHARI	CC Attack Potential
Likelihood	Attack Vector	-	-	-	Resources	-	-	Window of Opportunity
	Access Complexity	-	-	Contact Frequency	Activity	-	-	-
		Adversary Targeting	-	-	Activity	-	-	Knowledge of TOE
	Privilege Required	-	-	Difficulty	Resources	-	-	-
	User Interaction	-	-	Difficulty	Resources	-	-	-
	-	-	-	-	-	-	-	Elapsed Time
	-	Adversary Capability	-	Threat Capability	Resources	-	-	Expertise
	-	-	-	Probability of Action	Resources	-	-	Equipment
	-	-	-	-	-	-	-	-
	-	Adversary Intent	-	Probability of Action	Motivation	-	-	-
	-	Range of Effects	-	-	-	-	-	-
Impact	Integrity, Confidentiality, Availability	Vulnerability Severity	Reputation & Customer Confidence	Reputation	-	Integrity, Confidentiality, Availability	Integrity, Confidentiality, Availability	-
			Financial	Response, Replacement, Competitive advantage				
			Productivity	Productivity				
			Safety & Health	-				
			Fines & Legal Penalties	Fines and judgements				

의 계좌 정보를 탈취하는데 이용될 수 있다. 따라서 B1과 B96은 입력 데이터를 도청하여 암호화폐를 탈취하거나 사용자 정보를 탈취하는데 동일하게 사용될 수 있다. 그러므로 B1과 B96은 결합하여 하나의 노드로 표현될 수 있으며 오른쪽 베이즈 네트워크를 보면 하나의 노드로 결합된 것을 확인할 수 있다. 이처럼 서로 공유하는 노드들을 연결하여 각 공격트리를 결합함으로써 베이즈 네트워크에서 각 공격 목표에 대해 통합된 전체 위험성을 측정할 수 있다.

3.2.2 위협 노드의 사전 확률 계산

베이즈 네트워크를 계산하기 위해서는 루트 노드인 각 위협 노드들이 발생할 사전 확률(prior probability)을 계산해야 한다. 일반적으로 사전 확률은 통계 데이터를 기반으로 계산하거나 또는 전문가의 전문 지식을 바탕으로 주관적인 판단에 의해 설정한다. 암호화폐 지갑에 대한 위협의 통계자료는 아직 충분하지 않은 상태이기 때문에 우리는 각 위협의 사전 확률을 계산하기 위해 다양한 표준들에서 공격 발생 가능성(likelihood)을 계산하는데 사용되는 여러 가지 측정 요소들을 활용한다.

우리는 이전에 조사한 12개의 표준들 가운데에서 위험성 측정을 위한 기준 요소들을 제공하는 7개의 표준을 선택하였다. 그리고 위험성 측정 표준은 아니지만 CC 평가에서 취약점의 공격 가능성을 계산하기 위해 사용되는 Attack Potential 항목을 추가하였으며, 그 결과는 Table 6에 나와 있다. Table 6의 Likelihood는 위협의 발생 가능성을 표현할 수 있는 다양한 기준 요소들이며, 공통된 의미를 갖는 요소들은 동일한 행에 표시하였다.

우리는 위협 발생 확률을 계산하기 위해 CVSS Exploitability 메트릭을 기본 메트릭으로 활용한다. SP 800-30, FAIR, EBIOS의 경우 Likelihood의 각 요소 값을 계산할 때 정해진 방법이 없기 때문에 관리자의 주관적인 판단으로 결정된다. CC의 Attack Potential은 각 메트릭 값을 계산하기 위한 기준을 제공하긴 하지만 Attack Potential의 계산된 점수는 공격의 발생 가능성을 나타내는 것이 아니라 오히려 반대로 공격에 대한 저항성을 나타내는 값이기 때문에 적합하지 않다. 반면 CVSS의 Exploitability 메트릭은 위협이 발생할 가능성을 나타내며, 각 값을 계산하기 위한 기준이 마련돼 있다. 게다가 각 메트릭 값은 [0, 1]의 값을 갖기 때문에 이를 확률값으로 가정했을 때 결합 확률(joint probability)을 계산하는데 적합하다. 이와 비슷하게 N. Poolsappasit 등 [25]는 베이즈 공격 그래프를 이용한 위험성 관리 방법을 제시하였는데, 이 때 각 노드의 조건부 확률을 계산하기 위해 CVSS 베이스 메트릭을 활용하였다. 그리고 H. Zhang 등 [26]은 [25]의 방법에서 CVSS v2.0 메트릭을 사용했던 것을 CVSS v3.0 메트릭으로 바

Table 7. CVSS Exploitability metrics and the appended metrics with metric values.

Metric		Metric Value				
CVSS Exploitability metric	Attack Vector (AV)	Network (N)	Adjacent (A)	Local (L)	Physical (P)	
		0.85	0.62	0.55	0.2	
	Access Complexity (AC)	Low (L)	Medium (M)	High (H)	-	
		0.77	0.62	0.44		
	Privilege Required (PR)	None (N)	Low (L)	High (H)	-	
		0.85	0.62	0.27		
	User Interaction (UI)	None (N)	Required (R)	-	-	
		0.85	0.62			
Appended metric	Time Complexity (TC)	None (N)	〈= 6 m (M)	〈= 5 y (H)	〈= 10 y (E)	〉 10 y (X)
		0.85	0.78	0.42	0.05	0
	Expertise (EX)	Layman (L)	Proficient (P)	Expert (E)	Multiple Experts (M)	
		0.85	0.53	0.39	0.33	
	Equipment (EQ)	Standard (S)	Specialized (P)	Bespoke (B)	Multiple Bespoke (M)	
		0.85	0.47	0.35	0.30	

꾸어 각 위협의 조건부 확률을 계산하는 방법을 제시하였다. 우리는 [26]과 비슷한 방법으로 CVSS v3.1 Exploitability 메트릭을 기본으로 하고 Table 6을 바탕으로 추가된 측정 요소들을 이용하여 각 위협의 사전 확률을 계산한다.

Table 7은 위협의 사전 확률 계산을 위한 CVSS 메트릭과 메트릭 계산값을 보여준다. 이 때 Access Complexity의 경우 좀 더 정밀한 계산을 위해 Medium 값을 추가하였다. 또한 Table 6을 바탕으로 위협 발생률을 계산할 때 필요한 세 가지 메트릭을 추가하였다. Time Complexity(TC) 메트릭은 공격 성공을 위해 얼마나 많은 시간 복잡도가 필요한지를 나타낸다. 그리고 Expertise(EX) 메트릭은 요구되는 공격자의 전문지식이나 역량 수준을 나타낸다. 그리고 Equipment(EQ)는 공격에 필요한 장비나 소프트웨어의 수준을 나타낸다.

우리는 각 위협 노드의 사전 확률 Pr(T)을 계산하기 위해 [26]의 확률 계산 공식에 TC, EX, EQ 메트릭을 추가하고 최대 확률을 1.0으로 설정하기 위해 상수를 3.44로 조정하여 식 (2)를 정의하였다.

$$\Pr(T) = 3.44 \times AV \times AC \times PR \times UI \times TC \times EX \times EQ \quad (2)$$

따라서 계산되는 최대 확률은 1에 가깝다.

$$\Pr(T) = 3.44 \times 0.85 \times 0.77 \times 0.85 \times 0.85 \times 0.85 \times 0.85 \times 0.85 \approx 1.00$$

그리고 계산되는 최소 확률은 0이다.

$$\Pr(T) = 3.44 \times 0.2 \times 0.44 \times 0.27 \times 0.62 \times 0 \times 0.33 \times 0.3 = 0.00$$

TC 메트릭에서 공격이 즉시 가능한 경우는 시간에 대한 제약이 없는 경우이므로 PR, UI에서와 동일한 가중치를 적용하여 0.85로 설정하였다. 그리고 무차별 대입 (brute-force) 공격과 같이 많은 공격 시간이 소요되는 경우에는 공격 시간에 따른 공격 성공 확률을 다음과 같이 계산하였다. 우선 사용자들이 지갑장치나 계좌를 변경하는 주기를 평균적으로 5년이라고 가정하고, 이 때 90%의 사람들이 9년 후에는 지갑 또는 계좌를 변경한다고 가정하였다.

그리고 이 확률분포는 정규분포를 따른다고 가정했을 때 Fig.5에서처럼 평균 m=5, 표준편차 σ=3.1을 갖는 정규분포곡선을 갖는다. 만약 나머지 메트릭의 제약이 없는 경우에 5년의 공격시간이 소요된다면 공격이 성공할 확률은 50%이다. 따라서 식 (2)를 통해 TC 메트릭값을 역으로 산출할 수 있다.

$$\Pr(T) = 3.44 \times AV(N) \times AC(L) \times PR(N) \times UI(N) \times TC \times EX(L) \times EQ(S) = 3.44 \times 0.85 \times 0.77 \times 0.85 \times 0.85 \times TC \times 0.85 \times 0.85 = 0.5$$

따라서 TC가 5년이 소요되는 경우에는 메트릭 값은 0.42를 갖게 된다. 이와 동일한 방법으로 나머지 메트릭 값도 계산하였다.

Expertise와 Equipment 메트릭은 Attack Potential을 참조하여 추가하였다. 우리는 각 메트릭 값을 계산하기 위해 Attack Potential에서 사용하는 점수를 활용하였다. Time Complexity 계산과 비슷하게 공격의 제약이 없는 가장 하위 기준에 대해서는 메트릭 값을 0.85로 설정하였다. 그리고 이 값을 기준으로 Attack Potential 점수의 비율에 따라 메트릭 값을 결정하였다. 이 때 각 메트릭의 최하위 기준의 점수가 0이므로 기준 간의 비율을 계산하기 위해 모든 값에 1을 더하였다. 그리고 나머지 메트릭의 제약이 없는 경우(최소 점수를 갖는 경우)에 계산된 Attack Potential 점수를 기준으로 위협 발생 확률 계산에 사용될 메트릭 값을 도출하였다. 예를 들어 Expertise의 Attack Potential

최소 점수는 Layman 기준에 대해 1점이다. 그리고 Attack Potential의 나머지 네 개의 메트릭이 모두 최솟값을 가질 때 Expertise가 Layman인 경우 점수는 5점이다. 그리고 이 점수를 CVSS 계산 방식의 메트릭 값으로 환산했을 때 0.85로 설정한다. 그리고 나머지 기준에 대해서는 Attack Potential 점수 비율에 따라 메트릭 값을 계산하였다. 이 때 Attack Potential 점수는 높을수록 공격 성공의 확률이 낮기 때문에 점수가 높아진 비율만큼 메트릭 값을 낮춰 계산하였다. 예를 들어 Expertise의 Proficient 기준의 점수는 8점이고 Layman 기준보다 점수가 1.6배 높으므로 위협 발생 확률을 나타내는 메트릭 값은 다음과 같이 0.53으로 설정된다.

$$\Pr(T) = 0.85 \times 5 \div 8 = 0.53$$

이와 동일한 방법으로 나머지 Expertise와 Equipment 메트릭 값을 계산하였으며, 이 값들은 Table 8에 나타나 있다.

우리는 식 (2)의 사전 확률 계산식을 간략화하기 위해 식 (3)의 새로운 함수 F를 정의하였다.

$$F(AV, AC, PR, UI, TC, EX, EQ) = 3.44 \times AV \times AC \times PR \times UI \times TC \times EX \times EQ \quad (3)$$

이렇게 완성된 Table 7과 식(3)을 바탕으로 Fig.4의 베이즈 네트워크에서 일부 위협 노드들의 사전 확률을 계산하면 다음과 같다. 이 때 지갑 시스템은 모바일 지갑으로 가정하였다.

$$\Pr(T1) = F(N, L, L, R, M, P, S) = 0.30$$
$$\Pr(T2) = F(L, L, N, R, M, P, S) = 0.27$$
$$\Pr(T3) = F(N, H, N, R, M, E, S) = 0.18$$
$$\Pr(T4) = F(P, H, N, N, N, L, S) = 0.13$$
$$\Pr(T5) = F(N, L, L, R, N, L, S) = 0.53$$
$$\Pr(T18) = F(P, H, N, N, M, E, P) = 0.03$$

노드 T1-T4는 키로거 악성코드를 설치하기 위한 위협들을 나타낸다. T1은 이메일, SMS 피싱(phising)과 같은 사회공학 기법을 사용하는 것으로서 원격으로 공격이 가능하며, 얼마든지 쉽게 공격이 재현 가능하고, 악성코드 다운로드를 위한 사용자의 상호작용이 필요하다. T2는 악성AP를 사용하는 것으로서 로컬네트워크에 악성AP를 설치하고, 해당 AP에 사용자가 접속하는 상호작용이 필요하며, 악성 AP를 운영하기 위한 약간의 전문지식이 필요하다. T3은 공급망 공격 방법을 사용하는 것으로서 원격으로 공격이 가능하고, 공격을 성공시키기 위해서는 공급망 시스템에 대한 많은 사전지식과, 상당한 수준의 전문지식이 필요하다. T4는 USB와 같은 이동식 저장매체를 사용하는 것으로서 물리적인 접근이 필요하며, 사용자가 기기를 사용하지 않는 특정 시점에만 제한적으로 공격이 가능하다. T5는 설치된 키로거를 통해 공격을 수행하는 것으로서 주로 키로거 악성코드는 백그라운드에서 동작하며 사용자의 입력정보를 원격으로 공격자에게 지속적으로 전송할 수 있다. T18은 오류 주입(fault injection) 공격과 같은 물리적 공격을 통해 사용자 인증을 우회하는 방법으로서 공격의 재현이 어렵고 공격을 성공하기 위해서는 일반적으로 수개월 이상의 시간이 소요되며 높은 수준의 전문지식과 특수 장비가 필요하다. 이렇게 계산된 각 위협 노드의 사전 확률은 Fig.4의 default prior 항목에 나타나 있다.

하지만 만약 Fig.4의 시스템에 특정 보안 컨트롤

Table 8. Attack Potential metrics and converted metric values for threat probability calculation.

Metric		Attack Potential value	Attack Potential score when other metrics have lowest values	Converted metric value
Expertise (EX)	Layman	1	5	0.85
	Proficient	4	8	0.53
	Expert	7	11	0.39
	Multiple Experts	9	13	0.33
Equipment (EQ)	Standard	1	5	0.85
	Specialized	5	9	0.47
	Bespoke	8	12	0.35
	Multiple Bespoke	10	14	0.30

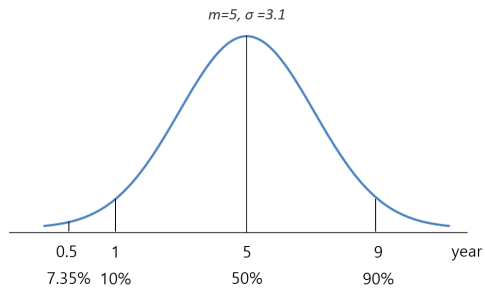

Fig. 5. Normal distribution curve of a cryptocurrency wallet change cycle.

(security control)을 적용한다면 각 위협의 발생 가능성이 달라질 것이다. 예를 들어 해당 장치를 외부 인터넷 네트워크와 분리하고 개인키를 secure element에 저장하여 물리적 보안을 강화하였다고 가정해보자. 이 때 장치가 외부 네트워크와 분리되었기 때문에 T1, T2 공격은 로컬네트워크로 제한되어 AV는 Local로 변경되며, 외부와의 접촉이 줄어들기 때문에 공격을 수행하기 위한 조건이 까다로워지므로 AC는 High로 변경된다. 그리고 T5 역시 악성코드를 이용해 데이터를 수집하기 위해서는 최소한 로컬 환경에 접근해야하기 때문에 AV는 Local로 변경된다. 그리고 secure element 사용으로 인해 물리적 보안이 강화됨에 따라 T18는 공격소요시간이 증가되어 TC는 Extreme으로 변경되고, 더욱 특화된 공격 장비가 필요하기 때문에 EQ는 Bespoke로 변경된다. 그리고 나머지 T2, T4에 대해서는 변경되는 사항이 없다. 따라서 보안 컨트롤이 적용된 후에 변경된 각 사전확률 결과는 다음과 같다. 이 값들은 Fig.4의 changed prior 항목에 나타나 있다.

$$\Pr(T1) = F(L, H, L, R, M, P, S) = 0.11$$
$$\Pr(T2) = F(L, L, N, R, M, P, S) = 0.27$$
$$\Pr(T3) = F(L, H, N, R, M, E, S) = 0.11$$
$$\Pr(T4) = F(P, H, N, N, N, L, S) = 0.13$$
$$\Pr(T5) = F(L, L, L, R, N, L, S) = 0.34$$
$$\Pr(T18) = F(P, H, N, N, E, E, B) = 0.00$$

변경된 사전 확률을 살펴보면 가장 변화가 큰 노드는 T1과 T5이다. 이것은 외부 네트워크와의 망분리 보안 컨트롤 적용에 의해서 악성코드를 설치하거나 또는 악성코드로 데이터를 탈취하는 공격들의 공격 노출면(attack surface)이 로컬 영역으로 제한되었기 때문이다. 그리고 노드 T18의 경우에는 secure element를 적용하기 전과 후의 사전 확률의 차이가 크지 않다. 이것은 일반적으로 물리적 공격 자체가 전문지식과 특수 장비를 요구하는 매우 까다롭고 어려운 공격 방법이기 때문에 기본적인 공격 발생 가능성이 낮기 때문이다.

3.2.3 하위 목표의 주변 확률 계산

우리는 각 지갑의 위험성을 측정하기 위해 공격트리의 각 하위 목표의 공격 발생 가능성을 계산한다. 이를 위해서 각 하위 목표를 도달하는 경로에 있는 모든 노드들의 조건부 확률을 이용해 결합 확률을 계산한다. 따라서 잘 알려진 식 (4)의 체인룰을 이용하여 각 하위 목표 노드의 결합 확률을 계산한다.

$$\Pr(x_1, \ldots, x_n) = \prod_{i=1}^{n} \Pr(x_i | Parent(x_i)) \quad (4)$$

그리고 식 (4)를 통해 얻어진 결합 확률들을 합산하여 해당 하위 목표 노드 S의 상태가 1일 때 $Pr(S=1)$ 주변 확률(marginal probability)을 계산한다. 이 때 각 노드의 AND, OR 연산에 대한 조건부 확률을 계산할 때는 Fig.3의 CPT를 사용하며, Fig.4에는 B10, B2, B7 등 몇 가지 노드들의 CPT가 나타나 있다. Fig.4에서 각 하위 목표 노드의 default marg과 changed marg 항목은 각각 시스템에 보안 컨트롤을 적용하기 전과 후의 해당 하위 목표의 주변 확률을 나타낸다. 예를 들어 노드 S1은 보안 컨트롤을 적용한 후에 주변 확률이 0.42에서 0.27로 낮아진 것을 알 수 있다.

3.2.4 공격목표의 위험성 측정

Table 9. Impact calculation scale and values.

Impact	Insignificant	Minor	Moderate	Major	Catastrophic
Financial	1	2	3	4	5
Reputation	1	2	3	4	5

우리는 각 하위 목표의 계산된 주변 확률을 바탕으로 식 (1)을 이용하여 각 목표에 대한 위험성을 측정한다. 식 (1)에서 Probability는 각 지갑별로 계산된 하위 목표의 주변 확률이다. 그리고 Impact는 해당 하위 목표가 사용자에게 끼치는 피해의 규모이다. 이 때 각 공격 목표에 대한 Impact를 계산하기 위해 Table 6의 Impact 항목 중 Financial과 Reputation 요소를 활용한다. Integrity, Confidentiality, Availability 요소는 암호화폐 지갑의 특성상 금전적인 피해 등을 표현하기에는 적합하지 않다. 또한 Productivity, Fines & Legal Penalties 등의 요소는 기업에게 해당되는 것으로 개인용 지갑에는 적합하지 않다. 따라서 Table 9와 같이 Financial과 Reputation 요소를 사용하여 Impact를 계산한다. 예를 들어 S1은 사용자의 개인키를 알아내 자산을 탈취하는 공격으로서 Financial은 Catastrophic(5)에 해당하고

Table 10. Cryptocurrency wallet security requirements checklist.

Domain	Category	Security Requirement	Impacted Node	Removed Node
(Omitted)				
Common	Output	a. Is there a mechanism to prevent screen capture when a private key or recovery phrase is displayed?	T10(AC, PR), T26(AC, PR), T35(AC, PR), T149(AC, PR), T202(AC, PR), T264(AC, PR), T312(AC, PR), T322(AC, PR),	
		b. Does the wallet deliver a warning message about the risk of exposing a private key or recovery phrase before they are displayed?	T11(AC), T27(AC), T150(AC)	
		c. Is user authentication required before displaying a private key or recovery phrase at the request of a user?		B10(X)
		d. Is there a mechanism to prevent screen capture when account or personal information is displayed?	T278(AC, PR)	
	Input	a. Is there a defense mechanism for keylogging attacks when a private key or recovery phrase is entered by a user?	T5(AC, PR), T26(AC, PR), T37(AC, PR), T149(AC, PR), T202(AC, PR), T264(AC, PR), T312(AC, PR), T317(AC, PR)	
	Copy	a. Is it forbidden to copy a private key or recovery phrase to the clipboard?	T36(AC, PR), T49(AC, PR)	
(Omitted)				
Embedded System	Authentication	a. Is there a mechanism for checking the authenticity of the wallet device that is connected to the host?	T71(PR, EX), T121(PR, EX), T172(PR, EX)	
	Authorization	a. Is there an authorization mechanism for the wallet manager that is installed on the host?	T55(PR, EX), T107(PR, EX), T159(PR, EX)	
Mobile	Privilege Escalation	a. Is there a mechanism to check if the device is rooted?		T40(O), T50(O), T66(O), T132(O), T154(O), T204(O), T229(O)

Reputation은 Minor(2)에 해당한다고 판단하여 Impact는 7로 설정한다. 이와 같은 방법으로 나머지 하위 목표들에 대해서도 Impact를 설정한다.

위험성을 계산할 때 최종 공격목표의 발생 확률이 아닌 하위 공격목표의 발생 확률을 계산하는 이유는 각 하위 공격목표별로 피해에 대한 Impact가 다르기 때문이다. 따라서 각 하위 공격목표별로 세분화하여 위험성을 각각 계산한 뒤, 이를 모두 합산하여 최종 공격목표의 위험성을 측정한다.

3.3 보안 요구사항 체크리스트

3.2.2장에서 설명한 것처럼 지갑 시스템에 적용된 보안 컨트롤에 따라 위협의 사전 확률이 달라진다. 따라서 실제 암호화폐 지갑들의 위험성을 평가할 때는 각 시스템에 적용된 보안 컨트롤에 따라 위협의 사전 확률을 다르게 계산해야 한다. 이를 위해서 3.1장의 위협 모델링을 바탕으로 보안 요구사항 체크리스트를 도출하여 Table 10에 작성하였다.

Table 10을 보면 도메인이 공통(Common) 영역과 특정 플랫폼 영역(Embedded System, Mobile)으로 나눠져 있다. 이것은 암호화폐 지갑에 공통적으로 필요한 보안 요구사항인지 아니면 특정 플랫폼 지갑에게만 필요한 보안 요구사항인지를 구분하기 위한 것이다. 그리고 체크리스트에서 Impacted Node는 해당 보안 요구사항에 따라 영향을 받는 노드를 나타낸다. 이 때 옆에 새겨진 (AC)와 같은 표시는 해당 보안 요구사항 만족 여부에 따라 변경되는 Table 7의 메트릭을 나타낸다. 그리고 Removed Node는 해당 보안 요구사항에 따라서 제거되는 노드를 나타낸다. 이 때 (O) 표시가 있는 것은 해당 보안 요구사항을 만족할 때 제거되는 노드를 의미한다. 반대로 (X)로 표시된 것은 해당 보안 요구사항을 만족하지 않았을 때 제거되는 노드를 의미한다.

IV. 암호화폐 지갑 위험성 평가 결과

본 논문에서 위험성 평가 대상이 된 암호화폐 지갑들은 충분히 많은 사용자를 확보하고 있으면서 해당 지갑의 구조나 설계에 대한 충분한 정보를 제공하는 참조 문서가 있거나 또는 오픈소스로 만들어진 지갑들이다. 우리는 콜드월렛 두 종류와 핫월렛 네 종류를 선정하였다. 콜드월렛은 전 세계적으로 가장 많

Table 11. Cryptocurrency wallet security requirements analysis result.

Domain	Category	Security Requirement	Ledger Nano S	Trezor One	Bread Wallet	Trust Wallet	Copay Wallet	Electrum Wallet
(Omitted)								
Common	Output	a. Is there a mechanism to prevent screen capture when a private key or recovery phrase is displayed?	O	O	X	O	X	X
		b. Does the wallet deliver a warning message about the risk of exposing a private key or recovery phrase before they are displayed?	X	O	X	O	O	O
		c. Is user authentication required before displaying a private key or recovery phrase at the request of a user?	O	O	O	O	O	O
	Input	a. Is there a defense mechanism for keylogging attacks when a private key or recovery phrase is entered?	O	O	X	X	X	X
	Copy	a. Is it forbidden to copy a private key or recovery phrase to the clipboard?	O	O	X	O	X	X
(Omitted)								
Embedded System	Authentication	a. Is there a mechanism for checking the authenticity of the wallet device that is connected to the host?	O	X	-	-	-	-
	Authorization	a. Is there an authorization mechanism for the wallet manager that is installed on the external host?	O	O	-	-	-	-
Mobile	Privilege Escalation	a. Is there a mechanism to check if the device is rooted?	-	-	O	O	-	-

이 판매된 Ledger Nano S와 Trezor One 하드웨어 지갑을 분석 대상으로 선정하였다. 그리고 핫월렛은 모바일 지갑인 Bread, Trust Wallet 지갑과 PC 지갑인 Copay, Electrum 지갑을 선정하여 분석하였다. 이 때 각 모바일 지갑과 PC 지갑의 운영체제는 전 세계에서 가장 많이 사용되는 안드로이드와 Windows 운영체제를 대상으로 분석하였다.

4.1 보안 요구사항 체크리스트 점검 결과

Table 11은 Table 10의 보안 요구사항 체크리스트를 기반으로 실제 시중의 지갑들의 보안 요구사항 만족 여부를 점검한 결과이다.

4.1.1 Ledger Nano S

Ledger Nano S 지갑은 하드웨어 지갑이며 secure element를 내장하고 있다. Ledger 지갑은 전체 지갑 중에서 가장 많은 보안 요구사항을 만족하는 것으로 나타났다. 기본적으로 임베디스 시스템의 특성과 네트워크 망분리로 인해 악성코드 감염 공격에 대해 안전한 것으로 나타났다. 특히 secure element를 내장하고 있기 때문에 물리적 공격 등에 안전하다. 하지만 Common 영역의 Output.b는 만족하지 않는 것으로 나타났는데, 이것은 개인키에 해당하는 복구문구를 사용자에게 보여줄 때 주의사항을 전달하는지 여부를 나타낸다. 따라서 복구문구 노출에 대한 위험성을 안내하지 않기 때문에 훔쳐보기 공격 등에 취약할 수 있다.

4.1.2 Trezor One

Trezor One 지갑은 하드웨어 지갑이며 범용 MCU를 내장하고 있다. Trezor 지갑 역시 기본적으로 임베디스 시스템의 특성과 네트워크 망분리로 인해 악성코드 감염 공격에 대해 안전한 것으로 나타났다. 그러나 Trezor 지갑은 일반적인 범용 MCU에 키를 저장하기 때문에 물리적 공격 등에 취약했다. 그리고 일정시간 지갑을 사용하지 않았을 때 자동 잠금 기능이 없기 때문에 자리를 비운 사이에 공격자가 접근하여 사용할 가능성이 있다.

4.1.3 Bread

Bread 지갑은 모바일 지갑으로서 스크린 캡처, 키로거, 클립보드 데이터 탈취 공격 등에 대한 보안 요구사항을 만족하지 않는 것으로 나타났다. 따라서 전반적으로 악성코드 공격에 취약한 것으로 확인되었으며, 자동 잠금 기능이 없어서 물리적인 접근 공격에도 취약한 것으로 나타났다. 하지만 개인키는 안드로이드의 Keystore 시스템에 저장되어 안전하게 관리되는 것으로 나타났다.

4.1.4 Trust Wallet

Trust 지갑은 모바일 지갑으로서 스크린 캡처와

Table 12. Risk measurement results of cryptocurrency wallets with three goals and eight sub-goals (P: marginal probability, I: impact, R: risk).

Wallet		G1. Stealing Cryptocurrency				G2. Denial of Service				G3. Privacy Breach			Total Risk
		S1	S2	S3	Sum	S4	S5	S6	Sum	S7	S8	Sum	
Ledger Nano S	P	0.37	0.32	0.63	-	0.51	0.76	0.39	-	0.88	0	-	-
	I	7	7	6	-	4	4	3	-	3	0	-	-
	R	2.59	2.24	3.78	8.61	2.04	3.04	1.17	6.25	2.64	0	2.64	17.5
Trezor One	P	0.4	0.35	0.63	-	0.55	0.77	0.39	-	0.89	0	-	-
	I	7	7	6	-	5	5	3	-	3	0	-	-
	R	2.8	2.45	3.78	9.03	2.75	3.85	1.17	7.77	2.67	0	2.67	19.47
Bread Wallet	P	0.93	0.41	0.57	-	0.58	0.82	0.39	-	0.89	0	-	-
	I	7	7	6	-	4	4	3	-	3	0	-	-
	R	6.51	2.87	3.42	12.8	2.32	3.28	1.17	6.77	2.67	0	2.67	22.24
Trust Wallet	P	0.84	0.35	0.57	-	0.57	0.82	0.39	-	0.84	0	-	-
	I	7	7	6	-	4	4	3	-	3	0	-	-
	R	5.88	2.45	3.42	11.75	2.28	3.28	1.17	6.73	2.52	0	2.52	21
Copay Wallet	P	0.92	0.4	0.56	-	0.57	0.81	0.39	-	0.95	0.92	-	-
	I	7	7	6	-	4	4	3	-	3	3	-	-
	R	6.44	2.8	3.36	12.6	2.28	3.24	1.17	6.69	2.85	1.84	4.69	24.9
Electrum Wallet	P	0.92	0.4	0.56	-	0.57	0.81	0.39	-	0.91	0	-	-
	I	7	7	6	-	4	4	3	-	3	0	-	-
	R	6.44	2.8	3.36	12.6	2.28	3.24	1.17	6.69	2.73	0	2.73	22.02

클립보드 데이터 탈취 공격을 방지하는 보안 컨트롤이 적용된 것으로 확인되었다. 따라서 악성코드 감염 공격에는 Bread 지갑 보다는 상대적으로 안전한 것으로 나타났다. 하지만 키로거 공격에 대한 보안 메커니즘은 없어서 여전히 입력 데이터 탈취에 대한 위협은 존재하는 것으로 나타났다. 그리고 지갑 자동 잠금 기능이 적용되어 있었으며, 개인키는 안드로이드의 Keystore 시스템에 저장되어 안전하게 관리되는 것으로 확인되었다.

4.1.5 Copay

Copay 지갑은 PC 지갑으로서 스크린 캡처, 키로거, 클립보드 데이터 탈취 공격 등에 대한 보안 요구사항을 만족하지 않는 것으로 나타났다. 따라서 전반적으로 악성코드 공격에 취약한 것으로 확인되었으며, 자동 잠금 기능이 없어서 물리적인 접근 공격에도 취약한 것으로 나타났다. 특히 개인키는 암호화되어 저장되기는 하지만 PC의 일반 저장 공간에 저장되기 때문에 악성코드 등에 의해 탈취되었을 경우 무차별 대입 공격 등에 취약하다. 또한 사용자 인증을 하지 않아도 암호화폐 계좌의 주소와 잔액을 알 수 있었으며, 계좌의 입출금에 대한 알림을 받기 위해 이메일 주소를 등록하는 기능이 있어 프라이버시 침해에 대한 위협이 존재하는 것으로 나타났다.

4.1.6 Electrum

Electrum 지갑은 PC 지갑으로서 Copay 지갑과 마찬가지로 스크린 캡처, 키로거, 클립보드 데이터 탈취 공격 등에 대한 보안 요구사항을 만족하지 않는 것으로 나타났다. 따라서 전반적으로 악성코드 공격에 취약한 것으로 확인되었으며, 자동 잠금 기능이 없어서 물리적인 접근 공격에도 취약한 것으로 나타났다. 그리고 개인키는 암호화되어 저장되기는 하지만 PC의 HDD, SSD와 같은 일반 저장 공간에 저장되기 때문에 악성코드 등에 의해 탈취되었을 경우 무차별 대입 공격에 취약한 것으로 나타났다.

4.2 위험성 측정 결과

우리는 3.2장에서 설명한 방법대로 3.1.3장에서 작성한 각 공격트리를 결합하여 베이즈 네트워크로 변환하고 Table 11의 보안 요구사항 체크리스트 점검 결과를 바탕으로 실제 암호화폐 지갑들의 위험성을 측정하였다. Table 12는 분석된 전체 6개의 암호화폐 지갑들의 위험성 측정 결과를 나타낸다.

첫 번째 공격 목표인 G1 암호화폐 탈취 목표의 위험성을 살펴보면, Nano Ledger S의 위험성이 가장 낮다. 그 다음으로 낮은 것은 Trezor One이다. 평균적으로는 하드웨어 지갑보다 소프트웨어 지갑의 G1에 대한 위험성이 1.41배 높게 나타났다. G1의 하위 목표 중에서 하드웨어 지갑과 소프트웨어 지갑의 위험성이 결정적으로 차이가 난 곳은

S1 개인키 탈취 하위 목표였다. S1을 달성하기 위한 가장 효과적인 방법은 악성코드를 이용한 공격이었다. 따라서 망분리되어 보관되는 임베디드 형태의 하드웨어 지갑은 악성코드 공격으로부터 비교적 안전하기 때문에 S1의 위험성에서 많은 차이가 발생했다. 그리고 S2의 위험성은 모든 지갑에서 S1에 비해 낮게 나타났다. 그 이유는 지갑을 이용해 공격자에게 암호화폐를 전송하기 위해서는 지갑에 물리적으로 접근하여 사용자 인증을 우회하거나 공급망 공격을 통해 변조된 지갑을 설치하는 방법 등이 있는데, 이러한 방법들은 기본적으로 공격 성공 확률이 낮기 때문에 위험성도 낮게 나타났다. 그리고 S3에 대해서는 하드웨어 지갑과 소프트웨어 지갑의 위험성이 비슷하게 측정되었다. 그 이유는 사용자의 암호화폐를 가로채기 위한 가장 효과적인 방법 중의 하나는 클립보드의 주소를 바꿔치기 하는 것인데 사용자가 하드웨어 지갑을 사용하더라도 주소를 복사할 때는 호스트의 지갑 매니저 프로그램을 사용한다. 따라서 하드웨어 지갑을 사용하더라도 호스트의 악성코드 감염에 대한 위험은 소프트웨어 지갑과 비슷하기 때문에 S3 위험성은 비슷하게 측정되었다.

하지만 G1에 대해서 secure element를 내장한 Ledger Nano S보다 일반 범용 MCU를 내장한 Trezor One의 위험성은 겨우 1.05배 높게 나타나 그 차이가 크지 않았다. 그 이유는 이전에 설명한 것처럼 암호화폐 탈취의 가장 효과적인 공격 방법은 악성코드 감염인데, 하드웨어 지갑 사용으로 인해 이미 위험성 높은 많은 위협이 제거되기 때문이다. 게다가 하드웨어 지갑에 접근하여 물리적인 공격을 통해 사용자 인증을 우회하거나 메모리에 저장된 키를 추출하는 공격은 secure element를 적용하지 않더라도 기본적으로 공격 성공 확률이 매우 낮다. 따라서 secure element를 사용하더라도 암호화폐 탈취에 대한 위험성의 감소 효과는 작은 것으로 나타났다.

두 번째 공격 목표인 G2 서비스 거부 공격 목표에 대해서는 Trezor One 지갑을 제외하고 전체적으로 비슷한 위험성이 측정되었는데, 그 이유는 지갑을 잠그거나 삭제하는 것은 지갑에 접근하는 것만으로 쉽게 가능하고 하드웨어 지갑을 관리하기 위한 지갑 매니저 어플리케이션은 일반 PC에 설치되기 때문에 악성코드를 이용한 DoS 공격이 가능하기 때문이다. 하지만 Trezor One의 G2 위험성은 유독 높게 나타났는데, 그 이유는 Trezor One은 사용자가 복구문구를 백업했는지를 검사하지 않기 때문이다. 만약 사용자가 복구문구를 어딘가에 적어놓지 않고 PIN을 잊어버리거나 지갑 자체를 잃어버렸을 경우 자산을 복구하는 것이 불가능하다. 따라서 Trezor One은 Table 11의 결과에 따라 S4, S5에 대한 Impact가 높게 설정되어 위험성이 증가하였다.

세 번째 공격 목표인 G3 프라이버시 침해 목표에 대해서는 Copay 지갑을 제외하고 모두 낮은 위험성이 나타났다. S7의 사용자 암호화폐 계좌 정보 탈취에 대한 위험성은 하드웨어 지갑과 소프트웨어 지갑에서 모두 비슷하게 나타났다. 그 이유는 하드웨어

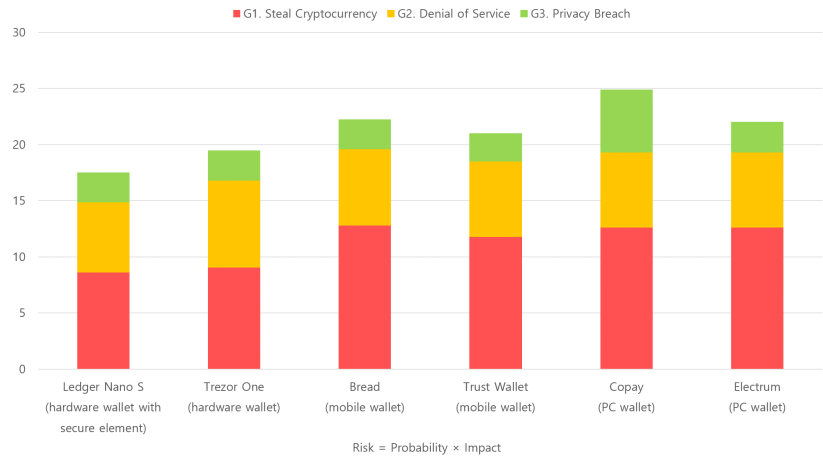

Fig. 6. Risk assessment results for 6 cryptocurrency wallets and three attack goals.

지갑을 사용하더라도 지갑 관리를 위해서 호스트에 지갑 매니저를 설치해야하기 때문에 계좌 정보 노출에 대한 위협은 소프트웨어 지갑과 비슷하게 나타났기 때문이다. 하지만 S8의 개인정보 탈취에 대한 위험성은 Copay 지갑을 제외하고 모두 0로 나타났다. 그 이유는 Copay 지갑을 제외한 모든 지갑들은 사용자의 식별가능한 개인정보를 입력받거나 저장하지 않기 때문에 S8 노드가 삭제되었기 때문이다. 반면에 Copay 지갑의 경우 암호화폐 계좌 정보에 대한 알람 설정을 위해 이메일을 입력하는 기능이 있어 S8 노드가 삭제되지 않았다. 따라서 G3에서는 Copay 지갑만 유독 높은 위험성이 측정되었다.

4.3 위험성 평가 결과

우리는 각 공격 목표별로 측정된 위험성을 모두 합산하여 각 지갑의 전체 위험성을 평가하였다. Fig.6은 Table 12의 합산된 전체 위험성 결과를 그래프로 나타낸 것이다.

전체 6개 지갑 중에서 Ledger Nano S 지갑이 17.5로 가장 낮은 위험성을 갖는 것으로 나타났으며, Trezor One이 19.47로 두 번째로 낮은 위험성을 갖는 것으로 나타났다. 따라서 하드웨어 지갑이 소프트웨어 지갑보다 전체적으로 안전한 것으로 나타났다. 평균적인 전체 위험성은 하드웨어 지갑보다 소프트웨어 지갑의 위험성이 1.22배 높게 나타났다.

소프트웨어 지갑 중에서는 Trust 지갑이 21의 가장 낮은 위험성을 갖는 것으로 나타났다. Trust 지갑은 스크린 캡처 방지, 루팅 여부 감지 등과 같은 보안 컨트롤이 적용되어 있어 전체 4개의 소프트웨어 지갑 중에서 가장 안전한 것으로 확인되었다. 반면에 Copay 지갑은 24.9로 전체 지갑 중에서 가장 높은 위험성을 갖는 것으로 나타났다. Copay 지갑은 사용자의 이메일 주소 등록으로 인한 G3의 위험성이 크게 증가한 것이 전체 위험성이 가장 높은 주원인이었다.

그리고 secure element를 내장한 Ledger Nano S보다 일반 범용 MCU를 내장한 Trezor One의 전체 위험성이 1.11배 높은 것으로 나타났다. 그런데 Trezor의 경우 복구문구 백업기능의 부재로 G2 목표에 대한 위험성이 유독 높게 측정되어 전체 위험성이 많이 증가했다. 따라서 secure element의 역할이 가장 중요한 G1 목표에 대해서는 Ledger 지갑보다 Trezor 지갑의 위험성이 1.05배 높았다는 점과 G3 목표에 대해서는 겨우 1.01배 높았다는 점을 고려했을 때 secure element를 사용하는 것은 지갑의 전체 위험성을 낮추는 데에는 효과가 미미한 것으로 확인되었다.

V. 결 론

우리는 위협 모델링을 통해 암호화폐 지갑의 보안 요구사항을 도출하고, 작성된 공격트리를 베이즈 네트워크 그래프로 변환하여 시중의 각 지갑의 위험성을 평가하였다. 평가 결과, 일반적으로 알려진 것처럼 하드웨어 지갑이 소프트웨어 지갑보다 안전한 것으로 나타났다. 그리고 하드웨어 지갑에서 secure element를 사용하는 것은 전체적인 위험성을 줄이는 데에는 효과가 낮은 것으로 나타났다. 그 이유는 일반적인 상황에서는 secure element를 사용하여 보호해야할 만큼의 심각한 위협이 발생할 가능성이 낮기 때문이다.

이 연구를 통해 우리가 도출한 암호화폐 지갑 보안 요구사항 체크리스트는 개발자들이 지갑을 개발할 때 참조하여 보안 내재화에 활용할 수 있다. 게다가 이 연구에서 제시한 방법을 통해 암호화폐 지갑의 위험성을 정량적으로 측정할 수 있기 때문에 특정 운용환경에 맞게 보안 컨트롤을 적용하여 위협의 Probability와 Impact를 조정하여 암호화폐 지갑의 위험 관리가 가능해진다.

References

[1] N. Satoshi, "Bitcoin: A peer-to-peer electronic cash system," https://bitcoin.org/bitcoin.pdf, Nov. 2008.

[2] G. Wood, "Ethereum: a secure decentralised generalised transaction ledger," https://ethereum.github.io/yellowpaper/paper.pdf, Dec. 2020.

[3] V. Buterin, "A next-generation smart contract and decentralized application platform," https://cryptorating.eu/whitepapers/Ethereum/Ethereum_white_paper.pdf, Apr. 2014.

[4] A. R. Sai, J. Buckley and A. Le Gear, "Privacy and security analysis of cryptocurrency mobile applications,"

2019 Fifth Conference on Mobile and Secure Services (MobiSecServ), pp. 1-6, Mar. 2019.
[5] Er-Rajy, L., et al., "Blockchain: bitcoin wallet cryptography security, challenges and countermeasures," Journal of Internet Banking and Commerce, vol. 22, no. 3, pp. 1-29, Dec. 2017.
[6] E. Almutairi and S. Al-Megren, "Usability and security analysis of the KeepKey wallet," 2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC), pp. 149-153, May. 2019.
[7] D. He et al., "Security analysis of cryptocurrency wallets in Android-based applications," IEEE Network, vol. 34, no. 6, pp. 114-119, Dec. 2020.
[8] M. Guri, "BeatCoin: leaking Private keys from air-gapped cryptocurrency wallets," 2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp. 1308-1316, Jul. 2018.
[9] Electric Coin Company, "Wallet app threat model," https://zcash.readthedocs.io/en/latest/rtd_pages/wallet_threat_model.html, Jul. 2019.
[10] SWhonix, "Cryptocurrency hardware wallet: threat model," https://www.whonix.org/wiki/Hardware_Wallet_Security, Dec. 2020.
[11] T. Hornby, "Invariant-centric threat modeling," https://github.com/defuse/ictm, Oct. 2019.
[12] M. Guri and Y. Elovici, "Bridgeware: the air-gap malware," Commun. ACM, vol. 61, no. 4, pp. 74 - 82, Mar. 2018.
[13] D. Nedospasov, T. Roth and J. Datko, "wallet.fail", 35th Chaos Communication Congress, https://wallet.fail, Dec. 2018.
[14] G. Miraje, M. Paulo and S. Leonel, "Trustzone-backed bitcoin wallet," Proc. The Fourth Workshop on Cryptography and Security in Computing Systems, pp. 25-28, Jan. 2017.
[15] W. Dai, J. Deng, Q. Wang, C. Cui, D. Zou and H. Jin, "SBLWT: a secure blockchain lightweight wallet based on Trustzone," in IEEE Access, vol. 6, pp. 40638-40648, Jul. 2018.
[16] Y. Liu et al., "An efficient method to enhance bitcoin wallet security," in 2017 11th IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID), Xiamen, pp. 26-29, Oct. 2017.
[17] P. Marek, et al., "BIP 39: Mnemonic code for generating deterministic keys." https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki, Sep. 2013.
[18] W. Pieter, "BIP 32: Hierarchical deterministic wallets," https://github.com/genjix/bips/blob/master/bip-0032.md, Feb. 2012.
[19] B. Schneier, "Attack trees," Dr. Dobb's journal, vol. 24, no. 12, pp. 21-29, Dec. 1999.
[20] B. Joachim, and N. Heninger, "Biased nonce sense: lattice attacks against weak ECDSA signatures in cryptocurrencies," in International Conference on Financial Cryptography and Data Security. Springer, Cham, pp. 3-20, Feb. 2019.
[21] J. Hoenicke, "Extracting the private key from a TREZOR," https://jochen-hoenicke.de/crypto/trezor-power-analysis, Nov. 2018.
[22] A. Bobbio, L. Portinale, M. Minichino, E. Ciancamerla, "Improving the analysis

of dependable systems by mapping fault trees into Bayesian Networks," in Reliability Engineering and System Safety, vol. 71, no. 3, pp. 249-260, Mar. 2001.
[23] Pappaterra, M. J. "Bayesian networks for online cybersecurity threat detection," Machine Intelligence and Big Data Analytics for Cybersecurity Applications, vol. 919, pp. 129-159, Dec. 2020.
[24] G. Marco, M. Iacono, and S. Marrone. "Exploiting bayesian networks for the analysis of combined attack trees," Electronic notes in theoretical computer science, vol. 310, pp. 91-111, Jan. 2015.
[25] N. Poolsappasit, R. Dewri and I. Ray, "Dynamic security risk management using bayesian attack graphs," in IEEE Transactions on Dependable and Secure Computing, vol. 9, no. 1, pp. 61-74, Jan. 2012.
[26] H. Zhang, F. Lou, Y. Fu and Z. Tian, "A conditional probability computation method for vulnerability exploitation based on CVSS," 2017 IEEE Second International Conference on Data Science in Cyberspace (DSC), pp. 238-241, Jun. 2017.

〈저자소개〉

유 병 철 (Byeongcheol Yoo) 정회원
2015년 8월~현재: ㈜키페어 수석연구원
2016년 2월: 가천대학교 전자공학과 학사
2021년 8월: 고려대학교 정보보호대학원 정보보호학과 석사
〈관심분야〉 보안공학 및 보안내재화 방법론, 블록체인, 암호화폐, IoT 보안

김 승 주 (Seungjoo Kim) 종신회원
1994년~1999년: 성균관대학교 정보공학과(학사, 석사, 박사)
1998년~2004년: 한국인터넷진흥원(KISA) 팀장
2004년~2011년: 성균관대학교 정보통신공학부 부교수
2004년~현재: 한국정보보호학회 이사
2011년~현재: 고려대학교 정보보호대학원 정교수
2014년~2015년: 육군사관학교 초빙교수
2014년~2016년: 다음카카오 프라이버시 정책 자문위원
2016년~2018년: 개인정보분쟁조정위원회 위원
2016년~현재: 산업통상자원부 전략물자기술 자문위원
2016년~현재: 한국카카오뱅크 정보보호부문 자문교수
2017년~현재: 고려대학교 국방RMF연구센터(AR2C) 센터장
2018년~2020년: 대통령직속 4차산업혁명위원회 위원
2018년~현재: 고려대학교 고신뢰 보안운영체제 연구센터(CHAOS) 센터장
2019년~현재: 중소벤처기업부 규제특례 심의위원
2020년: 합동참모본부 정책자문위원회 자문위원
2020년~현재: 해군발전자문위원회 자문위원
2020년~현재: 서울특별시 스마트도시위원회 위원
2021년~현재: 사이버작전사령부 자문위원
〈관심분야〉 보안공학 및 보안내재화 방법론, 자동차 및 무인이동체 보안성 평가 인증, RMF A&A, 암호학 및 블록체인