BGIN Meeting Report

Date: TBD
Location: Virtual Meeting

Executive Summary

The meeting focused on several critical aspects of blockchain security, including key protection, stakeholder perspectives, and the impact of regulatory frameworks. Key discussions centered around the need for a tiered approach to security, tailored to different use cases and levels of user assurance. Stakeholders, ranging from individual users to institutions and governments, were identified as crucial in defining these requirements. The importance of educating users about best security practices was emphasized, particularly in the context of managing keys and understanding regulatory impacts. Exchanges and custodial services were discussed for their role in managing user keys and providing security.

The meeting also highlighted the need for reference designs that define the lowest level of functional and assurance requirements. Enterprise vendors were noted as key contributors to operational insights and industry standards, with recent security incidents serving as case studies to underscore the importance of robust security measures. The session concluded with a plan to reconvene in 10 minutes to further define use cases and levels of assurance, acknowledging the valuable insights gained from the discussion. Overall, the meeting reinforced the significance of a comprehensive and collaborative approach to blockchain security, emphasizing education, stakeholder engagement, and industry best practices.

Key Discussion Points

1. **Security Evaluation and Key Protection in Blockchain Components**
   - Focus on the critical aspect of key protection within blockchain systems.
   - Emphasis on defining different use cases and their corresponding levels of assurance.
   - Importance of addressing security at various levels, including individual users, institutions, and governments.
2. **Stakeholder Perspectives**
   - Discussion of the diverse stakeholders involved in blockchain security, including:
     - Individual users who may have varying levels of technical understanding.
     - Institutions that require robust security measures for large-scale operations.

        - Governments that need to ensure regulatory compliance and protect national interests.
    - Challenges in aligning the needs and expectations of these different stakeholders.
3. **Challenges in User Education and Best Practices**
    - Highlighted the difficulties in educating users about security best practices.
    - Emphasis on the role of exchanges and custodial services in managing user keys and providing security guidance.
    - Importance of user education to prevent common security pitfalls.
4. **Role of Exchanges and Custodial Services**
    - Discussion on how exchanges and custodial services play a crucial role in managing user keys.
    - Impact of these services on user behavior, particularly in terms of security practices.
    - Need for clear communication and best practice guidelines from these entities to enhance user security.
5. **Importance of Reference Designs and Defining Use Cases**
    - Importance of creating reference designs that cater to different use cases.
    - Need to define the lowest level of requirements in terms of functions, robustness, and assurance.
    - Role of enterprise vendors in providing operational insights and contributing to industry standards.
6. **Impact of Regulatory Frameworks on User Behavior**
    - Discussion on how regulatory frameworks influence user behavior, particularly in relation to security practices.
    - Impact of regulations and taxes on the adoption of secure practices among users.
    - Need for a balanced approach between regulation and user convenience.
7. **Examples of Recent Security Incidents**
    - Review of recent security incidents and their implications.
    - Importance of learning from these incidents to improve security measures and best practices.
8. **Personhood Paper, World ID, and Personhood**
    - No specific discussion on the "Personhood Paper," "World ID," or related terms was mentioned in the detailed summary.
9. **Taxonomy of Harms, FASE, and Taxonomy**
    - No specific discussion on the "Taxonomy of Harms," "FASE," or related terms was mentioned in the detailed summary.
10. **Understanding is Key, Proverb, and Zcash**
    - No specific discussion on "Understanding is Key," proverbs, or Zcash was mentioned in the detailed summary.

Next Steps

- Define use cases and levels of assurance.
- Continue to gather insights from stakeholders.
- Plan to reconvene in 10 minutes to further discuss and refine these points.

Action Items and Next Steps

- **Immediate Actions**:
  - Define specific use cases and their expected levels of assurance.
  - Gather input from stakeholders, including individual users, institutions, and governments.
  - Develop reference designs for different use cases.
  - Educate users about best practices in key management and security.
- **Long-term Goals**:
  - Create a comprehensive framework for security evaluation in blockchain components.
  - Establish industry standards and guidelines for key protection and user education.
  - Foster collaboration between enterprise vendors, regulators, and the broader community.

Conclusion

- **Summary of Next Steps**:
  - Define use cases and levels of assurance.
  - Develop reference designs and best practices.
  - Educate users about security practices.
  - Reconvene in 10 minutes to continue discussions.
- **Final Thoughts**:
  - The session provided valuable insights into the challenges and opportunities in blockchain security.
  - Collaboration and a tiered approach are essential for addressing the diverse needs of different stakeholders.

Detailed Session Summary

24:56 - 57:04

- **Speaker**: [Discussion on security evaluation, key protection, and stakeholder perspectives]
  - **Key Points**:
    - Focus on key protection in the context of blockchain components.
    - Importance of defining different use cases and their expected levels of assurance.
    - Stakeholders include individual users, institutions, and governments.

        - Challenges in educating users about security practices.
        - Role of exchanges and custodial services in managing user keys.
        - Impact of regulations and taxes on user behavior.

57:04 - 1:29:24

- **Speaker**: [Continued discussion on use cases, requirements, and industry adoption]
    - **Key Points**:
        - Importance of reference designs for different use cases.
        - Need to define the lowest level of requirements in terms of functions, robustness, and assurance.
        - Role of enterprise vendors in providing operational insights.
        - Examples of recent security incidents and their implications.
        - Discussion on user education and best practices.
        - Impact of regulatory frameworks on user behavior.

1:29:24 - 1:35:04

- **Speaker**: [Wrapping up the session]
    - **Key Points**:
        - Summary of the discussion points.
        - Next steps for defining use cases and levels of assurance.
        - Acknowledgment of the insights gained.
        - Plan to reconvene in 10 minutes.

Content Analysis

- **Main Topics Discussed**:
    - Security evaluation and key protection in blockchain components.
    - Stakeholder perspectives, including individual users, institutions, and governments.
    - Challenges in user education and best practices.
    - Role of exchanges and custodial services.
    - Importance of reference designs and defining use cases.
    - Impact of regulatory frameworks on user behavior.
- **Key Insights**:
    - The need for a tiered approach to security based on the level of users and the value of assets.
    - Importance of educating users about security practices and best practices.
    - Role of enterprise vendors in providing operational insights and contributing to industry standards.
    - Examples of recent security incidents highlight the importance of robust security measures.

The meeting focused on several critical aspects of blockchain security, including key protection, stakeholder perspectives, and the impact of regulatory frameworks. Key discussions centered around the need for a tiered approach to security, tailored to different use cases and levels of user assurance. Stakeholders, ranging from individual users to institutions and governments, were identified as crucial in defining these requirements. The importance of educating users about best security practices was emphasized, particularly in the context of managing keys and understanding regulatory impacts. Exchanges and custodial services were discussed for their role in managing user keys and providing security.

The meeting also highlighted the need for reference designs that define the lowest level of functional and assurance requirements. Enterprise vendors were noted as key contributors to operational insights and industry standards, with recent security incidents serving as case studies to underscore the importance of robust security measures. The session concluded with a plan to reconvene in 10 minutes to further define use cases and levels of assurance, acknowledging the valuable insights gained from the discussion. Overall, the meeting reinforced the significance of a comprehensive and collaborative approach to blockchain security, emphasizing education, stakeholder engagement, and industry best practices.

1. **Security Evaluation and Key Protection in Blockchain Components**
   - Focus on the critical aspect of key protection within blockchain systems.
   - Emphasis on defining different use cases and their corresponding levels of assurance.
   - Importance of addressing security at various levels, including individual users, institutions, and governments.
2. **Stakeholder Perspectives**
   - Discussion of the diverse stakeholders involved in blockchain security, including:
     - Individual users who may have varying levels of technical understanding.
     - Institutions that require robust security measures for large-scale operations.
     - Governments that need to ensure regulatory compliance and protect national interests.
   - Challenges in aligning the needs and expectations of these different stakeholders.
3. **Challenges in User Education and Best Practices**
   - Highlighted the difficulties in educating users about security best practices.
   - Emphasis on the role of exchanges and custodial services in managing user keys and providing security guidance.
   - Importance of user education to prevent common security pitfalls.
4. **Role of Exchanges and Custodial Services**

- Discussion on how exchanges and custodial services play a crucial role in managing user keys.
- Impact of these services on user behavior, particularly in terms of security practices.
- Need for clear communication and best practice guidelines from these entities to enhance user security.

5. **Importance of Reference Designs and Defining Use Cases**
   - Importance of creating reference designs that cater to different use cases.
   - Need to define the lowest level of requirements in terms of functions, robustness, and assurance.
   - Role of enterprise vendors in providing operational insights and contributing to industry standards.
6. **Impact of Regulatory Frameworks on User Behavior**
   - Discussion on how regulatory frameworks influence user behavior, particularly in relation to security practices.
   - Impact of regulations and taxes on the adoption of secure practices among users.
   - Need for a balanced approach between regulation and user convenience.
7. **Examples of Recent Security Incidents**
   - Review of recent security incidents and their implications.
   - Importance of learning from these incidents to improve security measures and best practices.
8. **Personhood Paper, World ID, and Personhood**
   - No specific discussion on the "Personhood Paper," "World ID," or related terms was mentioned in the detailed summary.
9. **Taxonomy of Harms, FASE, and Taxonomy**
   - No specific discussion on the "Taxonomy of Harms," "FASE," or related terms was mentioned in the detailed summary.
10. **Understanding is Key, Proverb, and Zcash**
    - No specific discussion on "Understanding is Key," proverbs, or Zcash was mentioned in the detailed summary.

Next Steps

- Define use cases and levels of assurance.
- Continue to gather insights from stakeholders.
- Plan to reconvene in 10 minutes to further discuss and refine these points.

Appendix: Full Meeting Transcript (Anonymized)

```
47.7 --> 48.3:  Okay.
51.1 --> 51.9:  Let's start the session.
52.1 --> 59.2:  So, the session as introduced by Matsuo-san just before, the
```

session is on security evaluation,
59.2 --> 62.7: security target, protection profile topics.
63.6 --> 71.7: And to start with, we have an introduction to the concept of security evaluation by Yoshida-san.
71.8 --> 77.1: So maybe Hiro, you can start by introducing yourself before l aunching the presentation.
77.3 --> 77.9: Thank you.
78.0 --> 79.3: Thank you for your introduction.
79.9 --> 80.5: Can you hear me?
81.6 --> 82.1: Thank you.
82.1 --> 91.0: So, I'm working for a public research institute called AISD, and my institute is called Cyber
91.0 --> 93.9: Physical Security Research Center, CPSEC in short.
94.5 --> 99.3: And I'm also working for ISO SC27 Working Group 2.
99.5 --> 101.5: This is a crypto working group as a convener.
102.2 --> 108.9: And in Japan, I'm working for some sort of Japanese consort ium called IC System Security
112.5 --> 116.1: Consortium Roundtable as a security general.
124.2 --> 125.6: So this is the overview of the talk.
126.0 --> 130.7: First, I would like to give you some background information, such as security assurance and
130.7 --> 133.2: hardware security and IoT security.
133.2 --> 139.7: And afterwards, I will give a brief introduction to the sec urity requirement documents examples.
140.6 --> 150.3: The first one is CC-certified protection profile working on some sort of security IP called
150.3 --> 151.1: SCU.
151.5 --> 152.7: This is for IoT devices.
153.4 --> 160.3: And the second example is I would like to give you some exa mple introduction to this global
160.3 --> 164.5: platform security requirements called MCMPU profiles.
164.8 --> 165.4: And I will conclude.
169.5 --> 170.9: So security assurance.
172.9 --> 176.0: So I'd like to mention about the product level of security.
176.5 --> 179.8: So this is differentiated from algorithm level security.
181.0 --> 186.3: This level evaluation and certification are typically condu cted following internationally
186.3 --> 191.1: recognized common criteria, ISO IC 15408.
192.0 --> 198.2: So there are two key organizations, like evaluation facilit ies, ITSEF in short.
199.2 --> 204.9: And ICSEF assesses the security functionality of our produc ts, what we call target evaluation
204.9 --> 205.4: tool.
206.0 --> 209.9: To certify it meets the specific requirements.
209.9 --> 214.1: And the other organization is called a certification body, CB.
216.6 --> 221.8: The organization issues certificate for IoT products meetin g requirements of this standard.
222.6 --> 225.7: So here you have some examples of all these key organizatio ns.

226.6 --> 228.9: Of course, in Japan, we have IPA for CB.
229.7 --> 236.4: And one of the most large organizations is called, as ITSEF, is ACS BrightSight.
236.4 --> 241.0: And in France, you have ANSI, of course, and the U.S., you have MINIST.
250.8 --> 255.1: So let's take a look at what happens in the world in terms of security assurance.
256.1 --> 262.1: So of course, the objective is to provide secure products to the users.
262.8 --> 272.0: So if you go to Europe, in 2022, EU Cyber Security Act has been announced by the parliament.
272.7 --> 279.5: And now, if you want to do business in Europe, you have to comply with the security requirements.
280.8 --> 286.3: And what is interesting to me is that there are some mutual recognition arrangements between
286.3 --> 291.3: two schemes, a French scheme, CSPN, and a German scheme, BSZ.
292.5 --> 300.2: And in Singapore, you have a cybersecurity labeling scheme called CLS, certifying consumer
300.2 --> 301.2: IoT products.
302.6 --> 311.3: So what is interesting to me is that if you want to sell a Wi-Fi router, you have to comply
311.3 --> 312.6: with these schemes.
313.4 --> 314.2: So this is mandatory.
314.2 --> 318.3: In the U.S., you have a U.S. Cyber Threats Mark.
318.8 --> 327.2: And in Japan, METI, our ministry, announced in March last year that we have now new schemes,
327.3 --> 333.1: labeling schemes based on Japan cybersecurity technical assistance requirements, JCSTAR
333.1 --> 333.6: in short.
333.6 --> 344.4: So let me introduce our Japanese consortium called ICSRT, IC System Security Roundtable.
345.6 --> 354.4: So since 2009, we launched and we have focused on the IC chip security at the very bottom.
356.0 --> 362.5: For now, we have in Europe very strong organizations working on this smart card chip domain.
362.5 --> 364.1: It's called the EUCC ISAC.
364.8 --> 372.6: Now ICSSRT, subconsortium ICSJC, is now an associated member of EUCC ISAC.
372.7 --> 374.1: So we have some partnership.
375.2 --> 379.1: So nowadays, we are now facing IoT era.
379.5 --> 387.0: So we have many applications coming up, like automotive applications, industrial automations,
387.3 --> 389.4: industrial applications, smart energy, and so on.
389.4 --> 395.9: So we'd like to expand our scope from hardware security to the application security.
398.2 --> 404.7: So this is ICSRT, where my role is a security general.
406.7 --> 410.8: So let's take a look at a brief overview of the common criteria.
411.2 --> 412.6: So this is how it works.

412.6 --> 419.4: So in Japan, we have JISAC schemes, called Japan Informatio
n Security Evaluation and
419.4 --> 421.3: Certification Schemes.
421.6 --> 427.7: And what is nice about this scheme is this scheme is associ
ated with a third-party assessment
427.7 --> 429.2: in public certifications.
429.9 --> 436.0: So you have four players, procurers, applicant, and evaluat
ion facilities, and certification
436.0 --> 436.4: bodies.
436.4 --> 442.9: The procurers provide security requirements in the form of
a PP, a protection profile,
443.4 --> 444.7: security requirements specification.
445.3 --> 452.2: And then, manufacturer or vendors provide a security design
 specification in the form
452.2 --> 453.9: of a security target, NST.
454.9 --> 462.3: And then they submit to the evaluation facilities, and the
evaluation facilities evaluate based
462.3 --> 466.8: on what vendors submit, and make some evaluation reports.
467.6 --> 473.6: And the certification body's IPA validates and confirms a r
eport, if it's OK, and will
473.6 --> 474.6: issue a certificate.
480.3 --> 484.5: So now I'd like to briefly mention why the PP is important.
485.7 --> 491.2: So in this figure, you can see the secure development phase.
491.2 --> 498.6: So on the left bank, you have a development process on the
right hand, a test process.
499.2 --> 503.9: At the beginning of the development process, you have a des
ign phase, where you have to
503.9 --> 505.8: specify requirements.
506.2 --> 514.3: In terms of security, protection profiles, you must develop
 in order to specify the security
514.3 --> 515.0: requirements.
515.0 --> 518.7: So this is how the PP can be used.
524.6 --> 527.6: So the second background information is hardware security.
533.4 --> 539.2: So I'm talking about the smart card security, but the probl
em with the smart card is smart
539.2 --> 542.8: cards can move users anytime and anywhere.
543.8 --> 546.2: So the attackers can easily obtain them.
547.0 --> 554.5: So attackers may place a probable needle directly on the se
miconductor chip, and to obtain
554.5 --> 560.8: cryptographic keys, attacks may exploit their physical inte
rface by observing the power
560.8 --> 563.3: consumption or electronic magnetic waves.
564.7 --> 569.0: So in the field of hardware security, there are various kin
ds of attacks that have been
569.0 --> 571.3: already known since back in 2000.
572.3 --> 577.8: The European organization called SOGAIAS consolidated these
 attacks into 10 attacks
577.8 --> 582.8: categories, organized them, then established evaluation cri
teria.
591.7 --> 593.4: So what is important?

595.3 --> 601.1: It's very good, but sometimes people complain about the CC.
601.4 --> 603.8: Why? The CC is too generic.
604.6 --> 609.5: So if you want to apply to a specific business domain, technical domain, you have to develop
610.1 --> 611.0: specific schemes.
611.6 --> 617.6: So in terms of the hardware security, SOGAIAS is a leading, world-leading organization.
617.6 --> 623.8: So this is the structure of SOGAIAS in the Middle East.
624.3 --> 628.0: At the top you have JIWG, underneath you have three technical groups.
628.7 --> 632.4: J-HUST is the most important, relevant working group today.
633.3 --> 635.2: Jail, hardware, attack subgroup.
635.7 --> 640.5: This is a critical role in high assurance security variations, J-HUST.
641.5 --> 650.3: And sometimes J-HUST or SOGAIAS are producing some technical guidance for interpreting
650.3 --> 654.7: these documents and monitoring latest attacks and trends.
655.4 --> 659.4: It's a very important organization in terms of smart card or hardware security.
662.5 --> 666.9: So there has been something very big going on in recent yeas.
666.9 --> 672.0: Back in 2019, the EUC Cyber Security Act was adopted by the parliament.
674.0 --> 681.7: So there will be a transition from SOGAIAS to ENISA, standing for the European Union
681.7 --> 683.2: Agency for Cyber Security.
684.5 --> 689.7: It was tasked with establishing EU cyber security certification schemes on common
689.7 --> 691.6: criteria called EUCC.
691.6 --> 702.6: So the EUCC objective is to introduce unified European standards for IT products.
703.7 --> 708.0: But the important thing is that the smart card technical domain remains a focused
708.0 --> 710.0: technical domain at European levels.
710.4 --> 715.1: So if you look at the CSA, you notice that there are three levels.
715.6 --> 720.2: The green one, this is corresponding for government agencies.
720.8 --> 725.5: But what is interesting to me, it was new about the CSA, is in the middle, you see
725.5 --> 731.3: private certification is now made possible, where still third-party variation is
731.3 --> 738.3: required. But if your budget is quite limited, maybe you can apply this certification at
738.3 --> 739.1: the basic level.
743.0 --> 752.1: So CC methodologies are some very important, extremely important problems, which is how
752.1 --> 754.9: do you measure security against attacks?
755.3 --> 757.0: Measurement is very important.
758.2 --> 764.8: So SOGAIAS CC is addressing this problem by calculating wha

t is called the attack
764.8 --> 767.5: potential for smart card variations.
767.5 --> 769.8: So how do you measure it?
770.4 --> 780.5: So you have six security perspectives, how much time the at tack needs, how much, how
780.5 --> 783.2: deep the attacker's expertise can be.
783.7 --> 792.1: So from these six perspectives, you can calculate the value s and to sum up, you get
792.1 --> 795.7: the values of attack potentials.
795.7 --> 799.6: So this is ranging from zero to maybe 31 and above.
800.4 --> 808.9: So if your tool is a smart card, it is likely your point, y our values of the attack
808.9 --> 812.0: potential can be 31 and or above.
812.7 --> 819.2: And a smart card, similar devices, must resist attackers wi th a level of above five.
819.5 --> 821.8: So if you've got CC, you have five levels.
822.2 --> 824.9: The five is the highest level.
829.5 --> 838.0: So a little bit about a new trend, but from France, there i s some announcement from the
838.0 --> 845.1: CB, ANSI, about the transition from a conventional crypto t o the post-quantum.
846.8 --> 851.5: So what is interesting to me, there have been the ANSI, the French CB, is already
851.5 --> 855.1: issued two certificates on products on having PQC.
855.7 --> 860.2: Both of which are achieving the highest level of above five.
864.0 --> 867.2: So another background information is IoT security.
872.0 --> 873.9: So what are the problems?
874.9 --> 880.6: Now we are given a very good certificate schemes like CC, w hy do we need another
880.6 --> 888.0: schemes? But CC is maybe chips, maybe this is a very generi c one, but now we are IoT
888.0 --> 895.8: era. So not only we have to work on not only the smart card chip, but also chips for
895.8 --> 899.6: embedded devices, chips for IoT devices, even for our vehic les.
901.6 --> 907.0: So the question here is how to balance assurance cost and a ssurance strictness.
907.0 --> 912.8: So if you have a lot of money, if you have a lot of budget, maybe we can achieve this
913.5 --> 917.6: very strict assurance security like a smart card assurance.
918.0 --> 923.5: But if your budget is quite limited, maybe you can sacrific e maybe assurance or
924.1 --> 929.8: strictness, but you can achieve a very efficient, reasonabl e security assurance.
930.2 --> 934.6: Maybe you might need more lightweight schemes than common c riteria.
934.6 --> 940.8: But common criteria could be overkill for what you need if you are IoT developers.
941.6 --> 945.6: So this is the problem we address in the era of IoT.
949.0 --> 950.9: So more problems.

952.1 --> 956.5: So on the right side, you have IoT architectures.
956.9 --> 962.1: On the top, you have servers, data centers and you have Internet and network devices in
962.1 --> 967.9: the middle. At the bottom, you have IoT end nodes, very poor inexpensive devices.
970.5 --> 978.6: So if you want to achieve security at the bottom IoT range, so what we have is MCU
978.6 --> 986.0: microcontrollers. But typically, MCUs is a wide range of MCUs in terms of memory, power,
986.2 --> 987.0: energy and so on.
988.0 --> 994.3: And on the other hand, the crypto still plays a very significant role in IoT security.
994.8 --> 1000.7: But the problem is that public key crypto could be expensive, requiring considerable
1000.7 --> 1002.2: implementation resources.
1003.1 --> 1010.3: So the question here is how to apply the public key crypto could be expensive to IoT end
1010.3 --> 1012.0: nodes. So this is the problem.
1012.0 --> 1021.8: So if you want to choose IoT end nodes microcontrollers, you have a lot of choices.
1022.6 --> 1028.6: Maybe you want to consider several perspectives like memory, security and cost and
1028.6 --> 1033.9: architecture, connection ports and Internet protocols and power.
1035.1 --> 1041.5: And different vendors already offer some portfolios of MCUs to meet the requirements
1041.5 --> 1044.0: from a variety of IoT applications.
1044.7 --> 1048.6: Now you have a lot of vendors, Atomel, Microchips, NXP, Infineon and so on.
1052.0 --> 1057.4: So now this is the main topics that I'd like to introduce some example of the security
1057.4 --> 1058.5: requirement documents.
1059.2 --> 1063.9: The first one is the protection profiles I've developed.
1064.9 --> 1070.8: This is a profile, a single chip microcontroller equipped with a secure cryptographic
1070.8 --> 1074.3: unit version 1.0 to 1.20.
1075.1 --> 1080.9: This PP is specifying requirements for single chip microcontrollers in embedded devices,
1081.0 --> 1081.8: IoT devices.
1082.7 --> 1086.7: And this PP was issued and certified in 2022 in Japan.
1086.7 --> 1093.7: And the core technology, which I explained in the next slide, is SCU, secure cryptographic
1093.7 --> 1100.9: unit. And target IoT devices include sensors, actuators, so very cheap, inexpensive
1100.9 --> 1103.1: devices. This is our target chip.
1104.6 --> 1111.3: So as I explained, we have five above one level, but this level is only above two, a
1111.3 --> 1112.4: reasonable level.
1112.4 --> 1120.4: So this is a core technology called SCU, secure cryptographic unit.

1121.1 --> 1125.3: So you can use this technology as a route of trust for Io
T era.
1125.8 --> 1130.6: And this technology is already published in a journal in
2021.
1133.1 --> 1140.4: So this technology, SCU, enables you to use public key ca
pability, even need a cheap,
1140.4 --> 1143.6: inexpensive IoT devices.
1145.2 --> 1151.9: So what is nice about this technology is called in the mi
ddle, you see the security
1151.9 --> 1156.1: platform. This is a built-in access monitoring mechanisms.
1157.1 --> 1163.1: So if you're an attacker coming from the application soft
ware, so you have to overcome
1163.1 --> 1167.9: the two gates consisting of software gate and hardware ga
te in order to access
1167.9 --> 1170.7: the cryptographic engine.
1172.8 --> 1174.6: So this is quite nice.
1175.3 --> 1180.1: Maybe this could be a reasonable obstacle for the attacke
rs to access the cryptographic
1180.1 --> 1183.6: engine. So this is a very nice technology.
1186.9 --> 1190.5: So this is a PPST group I've heard.
1191.6 --> 1197.0: So one of the biggest problems when you want to develop t
he protection profile is maybe
1197.0 --> 1201.5: the first or maybe one of the biggest problem is that to
define a TOE, target of
1201.5 --> 1202.8: evaluation. This is difficult.
1203.8 --> 1207.5: The blue line shows the physical boundary that TOE.
1207.8 --> 1210.8: So you have a non-volatile memory, volatile memories and
so on.
1211.2 --> 1219.6: But what is interesting to me is, so if you look at the a
pplication software, surrounded
1220.9 --> 1222.4: by a red line.
1223.1 --> 1226.7: So physically this is inside of the TOE application softw
are.
1227.0 --> 1235.7: And try to use the cryptographic engines, circled with a
blue line, and via a software
1235.7 --> 1237.2: gate and hardware gate.
1237.4 --> 1244.1: But what is interesting is that logically this applicatio
n software is out of TOE.
1244.1 --> 1246.1: This is a bit tricky, but interesting.
1248.4 --> 1253.7: So the next problem is what you want to address is what a
re the assets you want to
1253.7 --> 1254.1: protect?
1255.1 --> 1260.5: So for each asset, you have to consider which security pe
rspective you want to consider,
1260.8 --> 1264.1: like confidentiality, integrity and availability.
1264.9 --> 1273.7: So in our case, the assets are integrity of security serv
ices and confidentiality and
1273.7 --> 1275.2: integrity of user data.
1280.3 --> 1282.7: So this table shows threats.
1283.0 --> 1283.8: What are the threats?

1284.6 --> 1293.5: In RPP, you have six of them tampering with application s
oftware and or hardware software
1293.5 --> 1294.0: gate.
1294.8 --> 1299.2: Secondly, abusing the key switch import function.
1310.4 --> 1317.3: Thirdly, analyzing changes in the TOE power consumption a
nd physically probing the inside
1317.3 --> 1321.1: of the TOE and physically manipulating the inside of the
TOE.
1321.9 --> 1323.4: So these are threats we are considering.
1329.2 --> 1337.6: So to ensure against these threats, we have main security
 functions provided by TOE.
1337.9 --> 1338.9: We have six of them.
1339.9 --> 1346.3: Detecting and responding unauthorized use of the cryptogr
aphic functions, preventing
1346.3 --> 1352.6: unauthorized use of the leakage, verifying the integrity
of the software gate during
1352.6 --> 1359.8: startup, providing key switch whose confidentiality and i
ntegrity are protected by
1359.8 --> 1368.6: cryptography and importing key switch from external entit
ies and updating after verifying
1368.6 --> 1370.2: the application software.
1371.4 --> 1372.5: These are main functions.
1378.6 --> 1384.1: So now I would like to explain the very, very basic quest
ions.
1385.4 --> 1388.7: The very reason why RPP exists.
1389.8 --> 1395.9: So you have now already very important and widely, global
ly widely used application file
1395.9 --> 1399.7: existing, PP84 on smart card.
1399.7 --> 1403.8: So this is very, very, very important processing profile.
1405.3 --> 1412.1: But this PP addressing the attacker of the high attack po
tential, typically above five.
1412.4 --> 1413.6: They're very, very strong attackers.
1414.5 --> 1418.9: So this means you have to spend a lot of money in develop
ing your products to resist these
1418.9 --> 1419.6: attackers.
1421.1 --> 1428.9: On the other hand, RPP, we have our target is TOEs used f
or IoT edge nodes, sensors,
1428.9 --> 1430.9: actuators, very cheap devices.
1431.7 --> 1434.7: So asset value handle is very low, could be very low.
1435.3 --> 1440.2: So attackers could be just pleasant criminals showing off
 his technology.
1442.5 --> 1449.7: So that your TOE could resist performer attacker, maybe o
f the basic attack potential.
1451.2 --> 1458.4: In this table, maybe our RPP, the attacker could be above
 two or maybe three at most.
1458.9 --> 1462.7: But non-PP is very, very strong attackers you have to add
ress.
1463.1 --> 1467.7: You have to achieve this high assurance security for your
 smart card above five.
1468.2 --> 1469.5: So this is the difference.

1474.7 --> 1478.2: So now let's take a look at the second examples.
1478.2 --> 1491.6: So this is the global platform schemes called SESIP, Security Evaluation Standard for IoT
1492.4 --> 1498.0: Platforms, owned by Global Platforms, and in order to reduce the cost in security
1498.0 --> 1506.7: certifications. The scope would be hardware and software library, real-time OS and so
1506.7 --> 1512.3: on. The assurance level ranging from one to SESIP 5.
1513.5 --> 1518.7: So the important thing is that it's already the methodology itself is already standardized at
1518.7 --> 1522.6: the European level as the EN 17927.
1524.0 --> 1528.7: So there are already 28 SESIP certified products.
1528.7 --> 1534.0: Out of them, you can see here certified chips from NXP.
1537.5 --> 1544.8: What is nice about this SESIP scheme is you can have only five packages available.
1544.9 --> 1545.9: This is very comprehensible.
1547.0 --> 1554.7: So you pick up some three security package here in the first examples and then achieve
1554.7 --> 1557.5: SESIP two assurance levels.
1558.7 --> 1560.1: So this is quite handy.
1565.3 --> 1571.2: So one of the most important documents developed by the global platform in terms of
1571.2 --> 1573.7: SESIP schemes is called SESIP methodology.
1574.4 --> 1583.0: This methodology document is quite equivalent to ISO 15408 standard in common
1583.0 --> 1586.6: criteria. It is a very fundamental document.
1588.7 --> 1593.8: And the SESIP focus is the security of the IoT things and the connected products based on
1593.8 --> 1595.7: this connected platform.
1596.4 --> 1600.9: So the yellow part, you have the connected products, the whole things.
1601.6 --> 1603.5: Underneath, you have connected applications.
1604.9 --> 1607.3: And then the most important one is the connected platforms.
1607.6 --> 1610.3: This is your tool in the CC world.
1611.7 --> 1614.0: This is an example of connected platforms.
1614.0 --> 1619.0: You have a crypto library, isolations, firmware and hardware at the bottom.
1620.1 --> 1626.9: The typical major assets in a connected platform is that you have user data and data in
1626.9 --> 1630.2: transit and product identities and so on.
1634.6 --> 1640.2: So what is nice about this SESIP methodology is this methodology document is only one
1640.2 --> 1643.7: document which is already specifying the threat models at the three levels.
1644.8 --> 1646.7: The first one is the basic scenarios.
1646.9 --> 1648.8: This is the essential threat models.
1649.4 --> 1653.0: The attack phase only involves remote access.
1653.7 --> 1659.8: Maybe you might have the Mirai viruses back in 2016, I would say.

1660.8 --> 1668.7: So this scenario already addressed a primary IoT concerns, scalable attacks, exploiting
1668.7 --> 1671.0: remote connections to connected platforms.
1671.0 --> 1676.6: But if you want to achieve the higher security levels, maybe you are interested in that
1676.6 --> 1685.5: second scenario, when the platform is physically accessible to attackers.
1687.5 --> 1691.5: And even more, you have scenarios like extended scenario.
1691.9 --> 1694.3: Maybe you might consider untrusted software.
1700.5 --> 1708.1: So in terms of the PP, a global platform already produced very popular profile documents.
1708.2 --> 1712.9: It's called SESIP Profiles for Secure MCU MPU Profile.
1713.7 --> 1720.0: So one of the characteristics, what I like about this document is called, this is user
1720.0 --> 1723.0: friendly, reader friendly writing style.
1723.3 --> 1724.0: Easy to read.
1724.3 --> 1727.8: But this is a bit different from the common criteria document.
1729.3 --> 1731.7: So here is the table of contents.
1733.9 --> 1739.8: Mirai writes down, you see the total scope of representation examples.
1740.8 --> 1745.6: So connected applications in the above, in the below, connected platforms.
1745.6 --> 1754.8: And as I told you, you have five packages, base features, security services, software
1754.8 --> 1757.8: isolations, hardware protections and secure enclave.
1757.9 --> 1762.4: So you have five, only five secure packages.
1762.4 --> 1776.0: So let's focus on the one basic packages, which is base ASP security functional requirements.
1776.7 --> 1781.8: As a base, the platform fulfills the following security function requirements.
1782.7 --> 1786.8: The first one, it provides a unique identification of itself.
1786.8 --> 1794.1: Secondly, it ensures its authenticity and integrity during platform initialization.
1795.3 --> 1798.3: And thirdly, it can be updated to a newer version.
1800.1 --> 1808.1: Fourthly, ensuring the list of data is erased before the memory is reused.
1808.1 --> 1817.1: Then finally, you have only providing list of endpoints authenticated with debug functionality.
1818.1 --> 1822.3: So this is five security requirements included in the base package.
1824.4 --> 1827.5: So this is mandatory security requirements.
1827.5 --> 1839.1: So as a final topic, what is nice about the CESIP scheme is what they call, what they claim.
1840.6 --> 1849.0: Maybe you can reuse CESIP certificate in order to harmonize two EUCC schemes.
1849.0 --> 1868.3: On the right side, you have European EUCC schemes, where you have five above one level, one, two, three, four, five and from above one, one to three, each of which is consistent to the CESIP level one, two, three.

1869.1 --> 1878.9: So in this way, if you obtain CESIP two certificate on yo
ur chip, maybe this certificate or maybe certification activity is consistent
 to the CESIP level one, two, three.
1879.0 --> 1886.8: This could contribute to obtain another certificate, I me
an EUCC certificate.
1887.4 --> 1894.9: So this is a big station, but I'm not sure this is true o
r not, but this is their hope, maybe my hope.
1898.6 --> 1899.7: Okay, this is the conclusion.
1899.9 --> 1905.3: I gave you some introduction to security assurance, hardw
are security, IOC security.
1906.2 --> 1912.4: After us, I gave some introduction of the two security re
quirements documents.
1912.5 --> 1918.6: One is CC certified processing profile and a global platf
orm CESIP profile.
1919.6 --> 1921.2: Okay, thank you very much for your attention.
1927.3 --> 1929.1: Any questions or comments?
1929.6 --> 1929.8: Welcome.
1935.3 --> 1937.0: Either online or in the room.
1945.7 --> 1948.1: Okay, I see no questions.
1948.5 --> 1950.2: Thank you very much for the introduction.
1950.8 --> 1951.0: Thank you.
1951.3 --> 1952.0: Overview of your work.
1955.1 --> 1963.8: Maybe for some it was very clear, maybe for some other it
 was a bit difficult because it's a topic which is not that easy.
1963.8 --> 1969.2: I mean, there are quite some complexity in the security e
valuation process, but I think you made a good overview.
1969.6 --> 1969.9: Thank you.
1970.6 --> 1971.3: Thank you.
1973.9 --> 1988.4: With this, the rest of the session is to discuss a bit wh
at we can do on that field within Begin for blockchain applications.
1990.0 --> 1994.8: But maybe before brainstorming, I would like to show a fe
w documents.
1996.6 --> 2009.0: There have been some initiatives in France around trying
 to think about some models for security evaluation of some blockchain compone
nts.
2011.3 --> 2018.2: The first one is coming from one major player in hardware
 wallet in crypto, Ledger.
2019.4 --> 2022.3: So I will show a few documents.
2037.7 --> 2044.4: So, Ledger is using chips within their wallet.
2044.5 --> 2052.0: They are using chips that have been security certified to
 some levels.
2054.7 --> 2076.7: As Hirotaka mentioned, some of the chips that are on the
market developed by the major players are evaluated for, by default, they are
 evaluated for being ready to be used, for instance, for ID documents or for
payment smart cards at the highest levels.
2077.7 --> 2091.3: So what Ledger did is reusing one of those chips and then
 building a target of evaluation.
2091.3 --> 2106.7: It's not really based on protection profile, but building
 a target, the security target, defining the target of evaluation that relies
 on security features provided by the

2107.4 --> 2116.6: chips they use and adding some security functions that they want to be able to demonstrate with respect to their wallet.
2118.7 --> 2123.0: So what they use, I'm trying to find one example.
2123.2 --> 2126.0: I think the first time they did it was in 2019.
2127.5 --> 2129.3: They have a few products.
2129.9 --> 2133.4: One is, I think this one will work.
2136.7 --> 2146.7: One, well, a few security variations that were made public are quite recent.
2146.8 --> 2166.3: The thing is that also when you go through, just to mention, when you go through a security evaluation process, in particular through those governmental, at government level, those assessed by the certification body at a country level.
2166.7 --> 2168.4: Those evaluations are public.
2168.8 --> 2187.2: Well, not all the details, of course, but your security target, the certification schemes that you are following, like whether this is Common Criteria 1, whether this is CSIP 1, or whether this is another one like CSPN, what you mentioned.
2187.7 --> 2189.2: This is something that is public.
2189.2 --> 2201.5: And then the report of the evaluation is also public without all the details, but you have the report showing at least what has been evaluated and what is the outcome of the evaluation.
2201.7 --> 2206.0: So just from that, you learn quite a lot already.
2207.0 --> 2216.7: So I just opened one randomly without any prior idea really of what is inside.
2216.7 --> 2221.7: I was just remembering this morning that we can display this in particular.
2221.9 --> 2222.8: It has been made in English.
2222.9 --> 2234.7: I guess it's more for the marketing purpose that it has been made in English because it has been evaluated in France by ANSI, but I guess it can be reused for marketing reasons.
2237.0 --> 2239.2: So normally it will come.
2240.9 --> 2245.7: So this seems to be the Ledger Nano X product.
2245.7 --> 2252.5: OK, so let's see what we have.
2252.8 --> 2261.2: So it's probably kind of extensive, but well, there is an introduction of the context.
2261.3 --> 2261.8: OK, fine.
2262.0 --> 2272.1: What you generally find in a security target is at least the product description, how the product is used.
2273.1 --> 2276.1: And then you have the target of evaluation.
2277.8 --> 2280.9: And with the definition of the target of evaluation, you have the threat.
2281.8 --> 2282.5: Well, it depends.
2282.9 --> 2294.0: But this one, my understanding is this one, this is not related to a protection profile, pre-existing protection profile, because it's based on the CSPN, the French schema, the French scheme.
2295.0 --> 2300.9: So you have to describe the threats that you want to consider.
2300.9 --> 2307.9: You have to describe the security function, well, basically the security requirements that you consider and the corresponding security function and so on.

2308.0 --> 2311.8: So you have to describe almost everything that you displa
y in your presentation.
2312.2 --> 2315.8: But here we have the target of evaluation, threat descrip
tion, security functions.
2315.9 --> 2316.7: What else do we have?
2317.3 --> 2329.8: And at the end, you have a mapping just showing that basi
cally your security functions are meaningful with respect to the assets and t
he threats that you want to avoid, the assets that you want to protect.
2330.9 --> 2334.0: And that everything is consistent.
2335.9 --> 2341.0: So I don't know if we look at the target of evaluation, h
ow they did this.
2341.7 --> 2344.4: I'm just not sure what we have.
2346.2 --> 2353.6: I don't know if there is some explanation of the meaning
of the course.
2363.4 --> 2365.4: I just discovered it right now.
2365.6 --> 2370.8: I remember that the first evaluation that they made in 20
19 was quite simple.
2370.9 --> 2378.5: It was really just a few software functions on top of the
 hardware they were using.
2378.8 --> 2383.6: This one seems to be more complete, at least at first gla
nce.
2383.6 --> 2394.6: So what I understand, at least from this figure, they are
 using a secure IC, which is coming from estimated microelectronics.
2395.2 --> 2406.5: This one, well, I don't know by heart for this, say, a nu
mber, but it's probably evaluated at Avavan 5 level following a common criter
ia scheme.
2408.2 --> 2412.5: And they are using a few components from this chip.
2412.5 --> 2415.7: In particular, they are using AES and AES accelerators.
2415.9 --> 2424.5: They are using a library, which is called Netscript, whic
h is quite well known in the field, which is providing also secure cryptograp
hic implementation.
2424.8 --> 2427.0: And they are using a random number generator.
2427.4 --> 2432.2: And we can guess that this is in particular important for
 the key generation process of the wallet.
2434.8 --> 2447.5: Then, on top of this, there is an embedded OS and some ap
plications in particular for the different blockchains that they are addressi
ng.
2448.0 --> 2450.9: In this one, we can see that they are mentioning Ethereum
 and Bitcoin.
2451.1 --> 2452.9: I don't know if they are considering the others.
2453.6 --> 2463.6: Here, in this evaluation, there is also a description on
the fact that there is a connection with the MCU that is in their wallet.
2464.3 --> 2476.1: From what I know, and again, I don't know necessarily all
 the details, but from what I know, they have the chip for more the crypto ke
y management and they have the MCU for the interface.
2476.1 --> 2491.3: And we can see the external world with the user, which is
 displayed because there is a USB and Bluetooth connection.
2492.2 --> 2503.7: So, what we have here, they say that the target of evalua
tion includes a few physical elements.
2504.7 --> 2512.0: So, buttons, screen, QRIC, and software by Piledriver.

2512.3 --> 2521.4: So, the firmware for the embedded OS on the chip and some embedded apps.
2521.7 --> 2532.7: And we can see here that, so, this is more complete than what I remember from the first security target that they had.
2532.7 --> 2555.7: So, they basically, from what I see, they are basically kind of describing, maybe there are some pieces missing, but kind of describing in the theory what correspond to the internal architecture of their wallet, plus some external physical elements like buttons, screen.
2556.9 --> 2564.3: And then, well, following, I mean, there are a few things in the description of the theories, including some assumptions.
2565.3 --> 2585.5: Where are they here? Well, yes, assuming, for instance, that this is an evaluation product that is considered as completely made and not modified, for instance, and not tampered beforehand.
2586.6 --> 2595.0: And, yeah, that's all. I can share the reference afterward. I can even share the link on the chat, by the way.
2597.4 --> 2599.1: Where is it? Here.
2604.6 --> 2605.3: Okay.
2609.5 --> 2619.1: Well, they are describing the assets within their TOE, so random number, secret seed, secret data protected by the PIN.
2623.0 --> 2636.0: PSD access control. So, PSD is the acronym for, I don't remember, what is it for, PSD?
2639.0 --> 2645.8: It might be physical security device, but, well, we can check if we want to discuss this.
2646.9 --> 2648.8: Embedded OS as well. Okay.
2649.8 --> 2652.3: Then they describe the threats. What are the kind of threats?
2654.7 --> 2660.9: Error in the wallet, weak random number generation.
2661.8 --> 2667.8: One threat is used by a user of non-legitimate ledger.
2669.9 --> 2671.2: Unwanted access.
2673.5 --> 2675.7: Compromising the post issuance capability.
2675.7 --> 2680.8: I guess this is related to their secure update process.
2681.9 --> 2683.2: Impersonation of the application.
2683.5 --> 2695.4: So, being able to mimic, for instance, a Bitcoin application in the wallet and then to use incorrect or malicious application.
2696.1 --> 2702.8: And then, with respect to that, they then define the security functions to run the number generator, attestation mechanism.
2704.4 --> 2706.0: What else do we have?
2707.0 --> 2710.8: End user verification. So, it's particular as a PIN entry.
2713.1 --> 2717.4: Post issuance capability. So, this is a secure update, I guess, again.
2719.9 --> 2721.3: What else do we have?
2722.7 --> 2723.8: Application isolation.
2724.7 --> 2737.8: So, I guess this is probably where they describe that you cannot interfere with an application by loading another application, by trusted application by loading another application.
2738.6 --> 2761.7: And then, just to describe, we are not trying to understand everything now, but just to describe that there are some aspects that are already known in the industry.
2761.7 --> 2778.3: Other projects that I'm, well, first of all, coming back to the hardware wallet, I don't know if anyone has some knowledge of other wa

llets that have been evaluated in the world.
2779.4 --> 2793.2: I've not tried to do an extensive review. As far as I kno w, I'm not aware of others, but maybe just that I've not looked deeply enough to the other provider.
2798.5 --> 2805.8: Then, we have, well, I think I prepared a few documents e lsewhere. Where are they?
2807.8 --> 2809.0: I don't remember.
2812.3 --> 2813.8: Maybe here.
2829.1 --> 2846.2: Another initiative that I am aware of, I think it's an ou tcome of a project, maybe a research European project or, I don't know, I thi nk a French one,
2846.2 --> 2853.5: organized in an excellent center called Cyber Security Ce nter in France.
2854.1 --> 2864.2: They work on trying to define a security relation target for an Ethereum node.
2865.2 --> 2880.0: Because when we think about security relation for blockch ain network, well, it could be wallets, could be nodes, could be a system ove rall, we'll have to define.
2881.0 --> 2907.3: This one is following the European standard 17640, which is, in fact, kind of similar to the CBSE lightweight version of Common Criter ia Approach.
2907.3 --> 2915.0: I would say it's kind of, and I say kind of between quote s, kind of equivalent of CESIP.
2915.7 --> 2917.9: But CESIP is really focusing on IoT.
2918.1 --> 2919.7: This one is a bit more general.
2920.7 --> 2933.2: But the way we call this standard in general in the field is fixed term evaluation.
2934.2 --> 2957.9: In the sense that the spirit of those lighter or lightwei ght security relation are to, well, the spirit, the principle are to reduce t he cost of security relation by limiting, constraining the number of days spe nt by the security evaluation laboratory on evaluating the target.
2957.9 --> 2974.2: Of course, the level of attacks that you are looking at c annot be as high as above and five that we have seen in the slide, for instan ce, but still trying to address the main threat.
2975.9 --> 2984.9: As an additional information, in general, in particular, this is true for the French schemes, CSPN, but I think this is also the same for 17640.
2987.1 --> 2991.6: It's for attacks up to the level of 11.3 in general.
2991.8 --> 2998.8: So we are trying to address the main general attacks and not necessarily the high end ones.
2999.4 --> 3003.9: So what they did, it's defining a security target for.
3006.8 --> 3009.1: For is to know.
3009.6 --> 3012.8: So let me just this one is in English as well.
3012.9 --> 3013.6: So that's good.
3014.7 --> 3016.9: I can share it here also.
3023.7 --> 3029.2: So maybe this one will be more complicated, but let me ch eck.
3030.8 --> 3033.0: At least we can see this is not a marketing.
3034.5 --> 3036.9: It's not meant to be marketing material.
3037.5 --> 3040.0: I don't know why it doesn't open again.

3040.2 --> 3041.2: OK, here.
3044.6 --> 3046.7: So what they described.
3050.0 --> 3056.7: Well, I don't know if there is a description of the theor
y that is clear somewhere.
3065.1 --> 3065.6: Well.
3066.8 --> 3074.7: Well, here is it clearly shows that the theory is one nod
e and it's only part of the overall blockchain.
3074.9 --> 3075.1: OK.
3076.1 --> 3077.2: And what do they say?
3078.5 --> 3087.9: Then we have a description of the theory, one node, which
 seems to include EVM.
3089.4 --> 3090.2: Oh, no.
3091.1 --> 3091.5: No, no.
3091.6 --> 3098.4: What is in black or dark is outside the theory.
3098.7 --> 3102.9: So it means that within the theory, we have some executio
n environment.
3104.3 --> 3109.9: The key store, what seems to be corresponding to decrypti
on.
3109.9 --> 3110.7: Decryption.
3113.6 --> 3118.0: Key for decryption and private key.
3119.7 --> 3122.9: The consensus execution software consensus algorithm.
3123.6 --> 3130.2: Just because there are things that are hard to control in
 such an environment.
3130.2 --> 3145.9: It's really beyond your control because it's part of the
theorem process that some aspects may evolve.
3147.2 --> 3154.0: So I don't see anything which is very easy to explain her
e.
3154.4 --> 3156.9: We have a description of the assets here.
3156.9 --> 3157.9: The funds.
3158.3 --> 3159.2: The smart contracts.
3160.5 --> 3161.2: Themselves.
3161.4 --> 3162.6: The user public key.
3162.8 --> 3163.5: Transactions.
3164.6 --> 3165.5: The source code.
3166.6 --> 3167.8: The node private key.
3167.9 --> 3169.4: The node public key.
3169.7 --> 3175.2: So all those assets are described and I guess somewhere t
here are the assumptions.
3177.6 --> 3184.0: Which is for instance to consider whether the server is i
n a secure environment.
3184.0 --> 3187.6: Whether it can be physically attacked.
3188.4 --> 3195.5: For other than three levels, you consider that the server
 is not going to suffer any physical attacks.
3195.8 --> 3197.5: But this is something that we may check if you like.
3197.9 --> 3199.1: And then there are the threats, etc.
3200.3 --> 3209.9: Let me show, before spending too much time on it, let me
show a last example of something that has been made also within the same kind
 of project.
3210.9 --> 3212.3: What is it here?
3213.5 --> 3216.9: There is another project.

3220.4 --> 3223.2: Secure target for smart contracts.
3228.5 --> 3229.2: Okay.
3229.9 --> 3232.7: This one is my project.
3232.7 --> 3233.2: Okay.
3239.4 --> 3241.0: I don't know, maybe not.
3241.5 --> 3242.5: Maybe this is in English.
3245.6 --> 3247.8: Okay, so everything is in English.
3247.9 --> 3248.2: Perfect.
3250.4 --> 3252.1: So the same principle.
3252.6 --> 3253.9: This is a secure target.
3254.3 --> 3256.4: So we'll have at some point a theory that is described.
3257.0 --> 3258.1: Here what they say.
3258.1 --> 3265.6: Target of evaluation is a smart contract implemented via ERC20 token standard for Ethereum and EVM compatible blockchains.
3265.7 --> 3274.0: So they looked at an ERC20 contract which is an open zeppelin template.
3275.7 --> 3278.1: Which is widely used for ERC20.
3281.2 --> 3283.0: What else do we know?
3283.0 --> 3291.0: So they explain the functional functions of the theory, so of this ERC20 smart contract basically.
3292.0 --> 3295.4: And they make some security assumptions.
3299.6 --> 3307.0: So some of the assumptions seem to be related to basically the blockchain itself.
3307.0 --> 3315.2: The underlying blockchain protocol shall be secure and particularly providing immutability of the smart contract once deployed.
3316.9 --> 3321.0: What else do we have?
3327.0 --> 3331.1: Here is a diagram of smart contract deployment.
3331.7 --> 3334.3: Do we have assets described somewhere?
3334.3 --> 3336.7: Or maybe I overlooked it.
3338.8 --> 3339.8: Target users.
3340.6 --> 3341.8: So this is the context of users.
3342.0 --> 3352.2: Again, this is based on CSP and it's a way to describe something which is lighter than going to a complex common criteria evaluation.
3353.5 --> 3355.5: Where are the assets?
3366.6 --> 3368.7: I cannot find them.
3370.9 --> 3371.5: Oh yeah, here.
3373.6 --> 3375.5: Contract by code, contract state data.
3376.3 --> 3377.5: Event logs.
3378.2 --> 3380.6: Gas token stored in the contract.
3381.3 --> 3383.4: Assets associated with functional assets.
3383.6 --> 3385.3: I'm not sure what it is.
3386.8 --> 3388.9: And then they describe the scope of evaluation.
3390.4 --> 3390.6: Wow.
3392.7 --> 3393.9: Here we have the assets.
3394.1 --> 3395.1: Contract by code.
3395.5 --> 3396.3: Event logs.
3396.5 --> 3397.2: User balances.
3397.6 --> 3398.4: User allowance.
3400.2 --> 3401.2: Token supply.
3401.6 --> 3402.2: Token name.

3402.3 --> 3403.0: Token import.
3403.9 --> 3404.7: And then assumptions.
3405.2 --> 3407.1: And then I guess we'll have the threats.
3407.9 --> 3408.3: And so on.
3409.3 --> 3410.6: So more to give.
3411.9 --> 3413.0: Did I share this one?
3413.1 --> 3413.6: I don't remember.
3415.0 --> 3416.4: Let me share this one on the chat.
3416.4 --> 3420.8: It's more to give a perspective of a few different initia
tives.
3421.2 --> 3423.5: Some are research projects.
3423.7 --> 3428.4: Although they are research projects, by the way, they try
 really to ensure the security.
3429.5 --> 3442.1: Or from what I remember, security in those research proje
cts were approved or at least ready to be approved by the certification body.
3442.1 --> 3448.6: In the sense that, well, you may write a security target
on your side.
3448.9 --> 3457.0: But if this is not something that either a security evalu
ation lab or a certification body is ready to approve,
3457.1 --> 3461.8: it means probably that the security target does not make
a lot of sense.
3461.8 --> 3469.1: So you need to be sure that it does the right level of in
formation for the different aspects that are expected.
3469.3 --> 3483.8: And then that it makes sense in terms of consistency betw
een threats, usage of your target evaluation, and the security functionalitie
s, security requirements that you are addressing.
3486.2 --> 3510.3: So, based on that, on this additional information, what d
o you think should be evaluated in a blockchain, in a blockchain context, in
a blockchain protocol or on Web3 applications or with respect to key manageme
nt?
3514.1 --> 3519.7: Any idea, any opinion, something that you want to try to
draft?
3525.8 --> 3526.2: Yeah, Eric?
3528.2 --> 3542.8: I'm a little concerned that, and I like the introductory
lecture on the five levels of secure, seven levels of security.
3545.8 --> 3552.9: Because, I mean, on the one hand, like the, and I've been
 using Ledger since 2019, so I'm glad they've been formally looked at.
3552.9 --> 3559.5: But I'm protecting a few thousand dollars worth of Bitcoi
n.
3560.4 --> 3580.2: So I don't need a HSM by bit level security, but someone
else might have a Ledger protecting $50 million worth of Bitcoin.
3581.1 --> 3594.1: So my concern is, you know, if we come out, you know, can
 we do anything other than here's like the absolute minimum requirements, or
maybe come out with a set of, because the answer is it kind of, it depends.
3595.0 --> 3596.5: And actually, that's part of it.
3596.6 --> 3601.4: Are we talking about consumer devices?
3601.5 --> 3603.6: Are we talking about the blockchain in general?
3604.1 --> 3612.8: And like that last one, where it looked like it was an ev
aluation of blockchain in general, that's got to be implementation specific,
no?

3614.4 --> 3621.7: You know, I'm building Ethereum in the Georgetown lab for fun versus doing a new DAO.
3622.2 --> 3624.5: And I would think that that would have different requirements.
3626.2 --> 3628.6: Yeah, I think this is a very good point.
3628.6 --> 3638.8: It's definitely dependent on use cases and slash what you do in practice, of course.
3638.8 --> 3640.3: But yeah, use cases.
3640.7 --> 3668.1: Those levels, for instance, in choosing to target other than five or other than four in quite early in the smart card industry was with the goal of producing electronic ID documents, banking card deployed widely, and really being sure that you don't have kind of reproducible attacks on.
3668.8 --> 3673.6: On those on those different documents and smart cards.
3673.8 --> 3691.7: When you think about crypto, well, depending on what you do, maybe, for instance, what you would like to show about CZIP with one of the levels sticking to considering remote attacks.
3692.3 --> 3694.9: Well, for some wallets might be enough.
3695.7 --> 3707.1: Well, well, you maybe you will have your, I don't know, $1,000 in your wallet stolen if someone grabs your wallet.
3707.3 --> 3710.6: But for many applications, remote attacks might be enough.
3711.1 --> 3716.0: While if you consider a secure custody service, that's a different story.
3716.1 --> 3717.1: Or an exchange, of course.
3722.5 --> 3724.1: Yeah, Daniel Saito.
3724.9 --> 3727.8: Yes, I like Eric's comment in regards to that.
3728.0 --> 3732.5: I fully agree in regards to the persona differences in regards to key management.
3732.5 --> 3749.9: You know, but I do believe that there should be at least a reference design and four different levels of security and be able to at least be able to say, because the attack vector and the attack surface is always there.
3749.9 --> 3765.9: And being able to educate the future user or the present day user and be able to say, well, the current day attack vectors, whether if you're using a software based crypto wallet to a ledger to even down to an HSM solution.
3766.0 --> 3777.6: I think it's good to be able to simplify it and be able to have a clear cut, whether if it's $1,000 or $1 billion on the wallet.
3778.0 --> 3780.2: It doesn't matter, security is security in that sense.
3780.7 --> 3782.7: And the security threats are increasing.
3783.2 --> 3785.4: They're getting more sophisticated in nature.
3786.1 --> 3791.0: And with that in mind, I think being able to have it easier to understand.
3791.0 --> 3795.9: I think default setting is to be as secure as possible, be as paranoid as possible.
3796.3 --> 3803.7: But of course, with those considerations, the difficulty level of adopting is always, you know, you get that user friction.
3804.4 --> 3807.2: And that's always a turnoff for any of the users.
3807.7 --> 3810.8: Like HSM solutions, a lot of user friction.
3811.0 --> 3813.3: But that friction allows for that security.
3813.3 --> 3822.7: And so as long as it's properly explained, as well as bei

ng able to say best practices would probably be a good idea.
3829.7 --> 3835.9: You know, I think of a Thales or Raytheon HSM, which is p
acked with thermite.
3836.8 --> 3841.4: So if you do try to open it up to put a probe in it, it b
urns.
3843.7 --> 3859.4: But someone who's, in fact, will go a few orders of magni
tude down, we're using some sort of future coin for, you know, tens of yen wo
rth of transactions.
3859.6 --> 3864.1: You're not going to spend even 100 yen on the card, righ
t?
3864.1 --> 3877.2: You know, we don't want to set the bar too high that it c
an't be used for low value transactions.
3877.6 --> 3889.9: But we also don't want to set it so low that if you are u
sing it for, you know, today's level of crypto transactions, that it's reason
able.
3889.9 --> 3891.8: You know, it's reasonable protection.
3892.3 --> 3897.1: And yeah, if it's a billion dollars, you have no problem
spending, you know, $30,000 on a Thales box.
3897.5 --> 3899.0: You're not going to carry it around. Yeah.
3905.8 --> 3911.0: Maybe there are two aspects that we can address.
3911.1 --> 3924.0: One, which is kind of trying to define or at least to rec
ommend some level of assurance, some level of assurance based on different us
e cases.
3924.8 --> 3926.4: That's one angle.
3926.4 --> 3942.0: The other one is, well, except some very specific, maybe
specific components in a blockchain protocol, they are all or almost all mana
ging keys in a way or in another.
3942.0 --> 3954.0: So for instance, trying to ensure that we can define at l
east the common threats that exist in most of the components and the common r
equirements.
3954.9 --> 3969.4: Like, well, there might be exceptions, but it's probably
true to say that if the key generation is flawed, then you are doomed in a wa
y or in another at some point.
3970.4 --> 3976.4: So key generation probably needs to have a certain level
of robustness.
3980.4 --> 3984.5: I can think of also some other common threats like remote
 attacks.
3985.3 --> 3992.6: Well, most of the cases, I think remote attacks will be a
 problem if they are possible.
3994.3 --> 3999.2: So maybe there is something related to access control, th
ose kind of things.
3999.2 --> 4015.5: So maybe we can have a definition of or at least trying t
o think about what are the common or basic aspects that needs to be taken int
o account versus levels of assurance on a use case basis.
4016.9 --> 4027.8: I'm trying to find the first ledger evaluation because I
think we will see some basic functionalities that were defined by that time t
hat could be helpful to look at.
4035.3 --> 4039.0: Am I hearing a proposal that we do work on?
4039.7 --> 4043.0: Maybe two or three profiles?
4051.6 --> 4060.0: Cheap, relatively, mostly secure, and banks, or maybe not

banks, because that's the thing.
4062.1 --> 4063.9: Somebody paid for all this, right?
4064.7 --> 4067.1: Banks already know what they're doing.
4067.1 --> 4074.8: I think maybe a focus on consumer and, I don't know, enterprises that are getting into the space.
4075.4 --> 4083.4: I would, well, banks are driven by compliance regulatory aspects, so they have to spend that money, and that's fine.
4083.8 --> 4089.9: And that framework is already kind of there in this document, and they could follow it to the T in that sense.
4089.9 --> 4096.8: And there's solutions that could be worth, you know, there's the IBM solutions, there's the Talos solutions, there's the Scirosa solutions.
4096.9 --> 4098.5: There's a list full of them.
4099.8 --> 4107.9: But the thing is that then you have the problem that we're seeing here in Japan is that we're seeing a lot of adoption at least kind of Web 3.0 levels.
4108.2 --> 4117.3: And they're seeing the news articles of exchanges getting hacked or wallets being drained in that sense, and it scares them.
4117.3 --> 4122.8: So they're really not getting into Web 3.0 mainly because of the security aspect.
4124.4 --> 4136.9: And if there is like a framework document of, you know, this is how you can get started, and this is at least the best practices of, you know, air gapping wallets.
4137.8 --> 4142.2: Not to have, just have good practices in key management.
4142.2 --> 4146.0: And, you know, no one person has all the keys or whatnot.
4146.1 --> 4151.0: If anything, they're all perhaps separated in that sense.
4151.2 --> 4156.1: And it requires a multi-sig to unlock the key management server in that sense.
4156.3 --> 4165.0: But, you know, yes, HSM is an overkill aspect, but I think one person having an overwhelming control over a wallet is also not a good thing as well.
4165.0 --> 4167.9: And it's a lot of responsibility for one person at the same time.
4167.9 --> 4179.1: So just being able to kind of design kind of the best practices, framework, as well as kind of the usability and kind of educate the user right now.
4179.2 --> 4183.4: We're in an interesting time where crypto is now becoming mainstream.
4184.3 --> 4188.5: A decade ago, people didn't know the difference between a public or a private key.
4188.8 --> 4190.2: Now this is actually mainstream.
4190.7 --> 4193.5: So it is a good time to, I believe, educate.
4193.5 --> 4202.8: And with that said, it is good to draft these type of documents now so that people can actually then now reference these designs and build on these.
4203.1 --> 4209.0: And so I think it's kind of our duty, too, to be able to write these.
4209.6 --> 4212.9: And I think we can own them and kind of guide the best practices.
4212.9 --> 4222.6: Because, look, the state-sponsored adversaries are gettin

g far more educated and they're faster, they're better.
4225.3 --> 4231.0: And if this continues, it's a cold war and it's actually taking place on blockchain.
4231.5 --> 4239.9: And the thing is that if we don't take care of this now, I believe that the adoption rate will just go slower and slower.
4239.9 --> 4245.9: And people will have that allergic knee-jerk reaction in regards to crypto.
4246.1 --> 4253.4: So I think we have the tenets to be responsible and be able to help drive that design.
4260.1 --> 4266.3: We think about potentially including governments and law enforcement as one of the personas as well.
4266.3 --> 4273.4: Obviously, recently, there have been seized funds that have been compromised in multiple jurisdictions.
4274.5 --> 4278.2: So I don't think it is necessarily something that has been solved yet.
4280.8 --> 4285.2: I'm glad I let you go first, because that actually doubles down on what I was thinking.
4285.2 --> 4294.2: One thing I noticed about both of the evaluation documents is they go through this detailed evaluation and basically say,
4294.3 --> 4295.9: and by the way, we promise you nothing.
4298.1 --> 4300.6: It's kind of like if we find a problem, yeah, it's a problem.
4300.7 --> 4304.3: But if we don't find a problem, it's still your problem.
4304.4 --> 4306.5: And by the way, we're only looking at half the system.
4310.4 --> 4315.9: And again, Bybit is kind of the poster child for, here are all the rules you follow.
4316.2 --> 4320.2: And they followed all the rules, and they lost a billion and a half dollars.
4323.6 --> 4326.1: But perfection is the enemy of the good.
4326.5 --> 4331.9: I think there's probably a lot of value of coming out with these guidelines.
4335.6 --> 4339.2: And let's face it, Ledger doesn't need our guidelines.
4339.9 --> 4343.5: But there are a lot of new companies popping up here and there.
4343.9 --> 4348.7: And as you said, a lot more consumers are starting to get attracted to this.
4349.0 --> 4360.0: And so having some baseline that gets people to say, oh, I ought to look and see what the latest research is and security.
4360.0 --> 4368.7: And maybe I'll spend $6 on a microcontroller instead of $3 on a microcontroller.
4369.3 --> 4373.6: But again, that's actually what I'm thinking, because when I bought the Ledger Nano, it was $59.
4374.4 --> 4377.0: Which means the manufacturing cost was under $10.
4377.6 --> 4386.0: Which meant if you had, in those days, MPC with a secure enclave, that probably cost $50.
4386.0 --> 4389.9: So it's already too expensive.
4395.0 --> 4395.9: Yeah, that's a good point.
4398.0 --> 4408.6: Big players, they already have processes to check compliance or to check whether they need additional assurance level for what they integrate.

4408.6 --> 4413.2: While they may still need some help to be sure that this is done properly.
4413.4 --> 4415.7: But that's a different story.
4415.8 --> 4422.4: In particular, when you think about Bybit or some others, it's not really about just one component.
4422.8 --> 4428.4: It's really about securing the overall processes, the overall system.
4428.9 --> 4437.2: And although this is something that might be feasible with those kind of approaches, formal security evaluation.
4437.2 --> 4439.7: This is going to be very costly.
4439.9 --> 4453.0: I mean, we try, for instance, to go through a formal verification of correct implementation of an OS, like Windows or some others.
4453.3 --> 4456.4: I mean, they invested a lot in order to do so.
4456.9 --> 4459.9: Just to be able to demonstrate that they are capable to do it.
4460.0 --> 4463.8: But it's a bit different goal.
4463.8 --> 4473.7: I like the fact that we may try to define baseline for those that are not really at the right level.
4474.8 --> 4475.6: Yeah, indeed Ledger.
4475.9 --> 4476.5: They already started.
4476.8 --> 4477.9: They don't need our help.
4478.3 --> 4482.1: Those providing HSM, well, normally not.
4482.5 --> 4486.6: I mean, you don't protect exactly against the same kind of attacks.
4490.1 --> 4496.1: Just still, I was thinking about trying to find the early evaluation of Ledger.
4496.4 --> 4498.6: I was not able because that's also...
4499.7 --> 4510.6: Well, at least I was not able to find it on the ANSI website because the evaluation they went through in 2019 is now expired.
4513.2 --> 4517.1: And they had another one in 2023, which is kind of related.
4517.5 --> 4523.2: And this one compared to the one I displayed before, the screen was not on it.
4523.3 --> 4524.7: The MCU neither.
4525.1 --> 4530.7: Also, they had an MCU to control and to drive the display.
4530.9 --> 4532.7: It was outside of the TOE.
4533.0 --> 4538.5: And the TOE was really on secure generation and access control of the user.
4538.5 --> 4548.8: And I think something related to the secure update of the chip, which makes sense within their context.
4549.7 --> 4565.3: But if you think about something which is a bit lighter in terms of complexity or cost for a wallet, it will bring us to something which is more secure generation.
4565.3 --> 4569.3: To be defined what we mean by secure and access control in a way.
4569.9 --> 4573.7: And probably something related to remote attacks if we want to think about it.
4573.9 --> 4576.3: But it's also a different, maybe a different story.
4576.8 --> 4577.2: Go ahead.
4578.3 --> 4589.7: From so-called Web3 users' perspective, I'm pretty much s

upportive of kind of scale of the security.
4590.1 --> 4594.4: It depends on the level of the users, amount of the money transferred.
4594.4 --> 4606.2: But as in the general users' perspective, it's very, let's say, in lesser portion about the security, level of the security.
4607.7 --> 4614.6: It only depends on the rumors or marketing messages from the providers.
4615.7 --> 4623.1: And yesterday we discussed about the information sharing amongst the providers of the services.
4623.1 --> 4642.2: But if our users who experienced some compromise of the security of the wallet I'm using, I don't know who report to, who can I share the information that experience.
4642.2 --> 4652.9: Or I may not be hacked, but just I lost, I was doing something wrong in just the processes.
4653.9 --> 4670.9: So I was just wondering if we were to generate some documents and guiding materials, how can we include some of the experiences of any kind of the general public,
4670.9 --> 4677.0: or use that kind of developed users who might have such experience?
4677.7 --> 4683.4: There could be some pool of information, but I don't know where it is.
4690.8 --> 4693.6: Well, that part is like, how do we set up the clearinghouse?
4693.6 --> 4706.5: You know, are we open to like a user sending us an email or going fill out a web form?
4706.6 --> 4713.6: Or do we say everything comes through whoever provides the software, whoever provides, in this case, the smart card or the wallet?
4718.0 --> 4721.0: Yeah, no idea. Anyone else have any idea?
4721.0 --> 4738.4: Yeah. And in traditional consumer products, there are several organizations who test such like a consumer board type of these things, or motor vehicles, anything.
4741.2 --> 4755.0: So you can start doing some document developing, but may you get to find some of the several such organizations if there is.
4759.3 --> 4766.5: Another model, but again, it's very high value, which justifies the cost.
4766.8 --> 4779.3: And the United States, Canada and Europe all have reporting systems for drug interactions, where adverse effects when you have sequelae from a drug.
4781.1 --> 4793.9: But it's, you know, a very high value, you know, the cost to society of realizing after a drug, you know, when it's in phase four, you can harm millions of people.
4794.1 --> 4796.4: So, you know, we'll pay lots of money for that.
4797.8 --> 4809.6: Is the financial services system, you know, would JFSA or CFTC or, you know, name your favorite local regulator?
4811.9 --> 4820.7: Because in a sense, though, because it's expensive, but it's spread out amongst everyone, that makes it affordable.
4821.7 --> 4829.0: Or do we just say, it's all Coinbase, and so, you know, Coinbase's customer service will ignore everything.
4829.2 --> 4831.1: I mean, take care of everything, we are recorded.
4832.5 --> 4836.1: They have very good, I've never used their customer servi

ce.
4836.2 --> 4839.4: I have no opinion on Coinbase's customer service.
4856.2 --> 4859.7: I think some examples could be taken from past hacks, for example.
4860.2 --> 4862.1: Let's say, I mean, Bybit's a great example.
4862.3 --> 4863.7: I mean, they played everything by the book.
4864.5 --> 4865.8: They had a multi-sig and everything.
4866.1 --> 4871.0: And if anything, it was just the upstream channel that was actually taken out, which is unique.
4871.3 --> 4872.6: It was a unique case.
4872.7 --> 4874.6: And could it have been avoided?
4875.4 --> 4875.8: Probably.
4877.0 --> 4878.8: And how could it have been mitigated?
4878.8 --> 4880.8: I think it's probably analyzing.
4881.3 --> 4883.8: Someone actually taking the time and looking at past hacks.
4884.7 --> 4890.4: I mean, there's plenty of them in Japan, and Bybit being one of the larger ones.
4891.1 --> 4898.3: I think it's analyzing it and, you know, kind of simplifying it to understand where it was attacked.
4898.5 --> 4899.6: I mean, it's well-documented.
4900.1 --> 4901.1: And we could start from there.
4901.2 --> 4902.3: We're backwards.
4902.3 --> 4915.6: And then how modern-day approaches in our reference design or best case, we can build comparisons of how it would have been mitigated in this sense.
4915.8 --> 4930.0: And so it's kind of a technical white paper, but yet more simpler for any CXO or CTO or CSO to understand what actually took place and how it could have been mitigated as post-forensics analysis or a report.
4931.9 --> 4939.7: But, you know, a lot of these companies are embarrassed to admit to their shortcomings or losses.
4940.3 --> 4946.2: But, you know, Bybit, they ran the master class in crisis management, and they did really well.
4946.6 --> 4947.8: A good job, I would say.
4950.5 --> 4953.4: And, you know, at least we learned something from this.
4953.4 --> 4963.4: And I think Bybit would want you to learn as, you know, all the enterprises or, you know, banking in the future from this example.
4964.0 --> 4971.6: And I think they would probably agree to that and perhaps, you know, own it and be able to support such endeavors.
4974.0 --> 4979.1: To the point that they'll fund some begin work for us to do something?
4979.9 --> 4981.1: You have a friend to call?
4985.7 --> 4992.7: I see different ways to proceed.
4994.6 --> 5012.0: Either we focus on more on key protection, and in that case, the topic of security evaluation with respect to common criteria, CSP and so on, makes sense.
5012.0 --> 5023.5: Or maybe, yeah, I like the idea to look at also at the recent acts, and Bybit is a very good example.
5024.5 --> 5035.5: And basically to see, well, first of all, I think it's a general trend for respect to different topics in begin, but not only in begin.

5035.5 --> 5048.6: There is definitely a need to explain at the right level what happened and to continue educating stakeholders, users on what happened.
5049.2 --> 5051.5: Although they did a good job already.
5053.4 --> 5058.8: And based on that, what was missing or what could have been done differently?
5060.0 --> 5074.1: Probably one aspect that we made occur is besides the overall governance processes, because it's not always about one step that failed.
5074.2 --> 5082.0: It's different layers, different controls that might have been helpful to catch something in advance.
5082.0 --> 5098.9: But something which is typical in Bybit and some other acts is lack of ways to validate, to check that the transaction signing request is really legitimate.
5098.9 --> 5117.2: So it's kind of only part of the issue for Bybit, but it's kind of narrowing a bit the scope to stand at the same level as the key generation problem or key protection problem.
5117.2 --> 5131.1: On the other side, there is one important functionality in many use cases, which is when do you decide and how do you decide to sign a transaction or not, and how you are able to check that you are signing the right transaction.
5140.6 --> 5144.9: This kind of security topic is new to me.
5144.9 --> 5152.5: So my question could be a beginner question, but I have the question.
5152.7 --> 5163.2: My question is, when it comes to security target for blockchain components, what kind of stakeholders are they for?
5164.2 --> 5174.8: So, I mean, like Dr. Boggess said that the customer has different perspective for security target.
5175.8 --> 5181.9: So, for example, in Japan, most of the crypto holders have low value of crypto.
5181.9 --> 5191.1: So when it comes to security target, who needs to think about it?
5191.8 --> 5200.2: So in my opinion, most of Japan's customers doesn't care about it.
5200.9 --> 5208.9: But an institution or government wants to protect the customer or wants to make the rules.
5208.9 --> 5216.4: So my question is, what kind of stakeholders are they for?
5222.8 --> 5227.4: And this highlights the schizophrenic nature of cryptocurrency.
5228.4 --> 5233.1: Because, oh, we want no institutions in the middle.
5233.3 --> 5235.1: We don't want companies involved.
5235.4 --> 5236.7: People are self-sovereign.
5236.8 --> 5238.3: They manage their own keys.
5238.4 --> 5239.4: They do all that.
5240.0 --> 5242.1: And they're incapable of doing it.
5242.2 --> 5243.9: So we have exchanges.
5243.9 --> 5248.3: We have not escrow wallets.
5248.4 --> 5252.1: What's the term where you hold the wallet for me?
5253.3 --> 5253.8: Pardon?
5255.2 --> 5255.7: Custodian.
5256.0 --> 5256.1: Yeah.

5256.7 --> 5257.5: We have custodial.
5259.1 --> 5262.9: So, and again, this is kind of back to the same story.
5263.0 --> 5264.7: We have a number of different answers.
5268.0 --> 5276.3: Honestly, it would be a great academic exercise to come up with the requirements for individuals to do everything themselves.
5276.5 --> 5278.3: But I don't know that the market's that big.
5279.0 --> 5291.0: Unless we can come up with something that's so easy to use and so inexpensive that you could have a truly distributed thing that somehow they'd, when their wallet.
5292.2 --> 5298.7: You know, again, if it's on a smart card and it goes into the laundry, they can recover their keys.
5298.9 --> 5301.0: But yet a hacker can't recover the keys.
5301.7 --> 5304.5: You know, all the things that normal people do.
5304.8 --> 5306.5: Not people like you and me.
5306.9 --> 5308.3: My ledger's in a safe.
5308.4 --> 5309.5: A fireproof safe.
5311.3 --> 5312.7: Yeah, that's not going to help.
5313.5 --> 5314.8: That's not your average user.
5315.8 --> 5318.6: And it's only, actually now it's only a few hundred dollars.
5318.6 --> 5320.4: I've unloaded most of my Bitcoin.
5321.7 --> 5326.4: So, for the case, to answer your question in regards to the Japanese user.
5327.5 --> 5340.6: Now, the majority of crypto, at least from a Japanese consumer perspective, we've noticed that a lot of the crypto holdings are actually stored with exchange.
5341.1 --> 5343.1: Which is kind of scary, actually.
5343.1 --> 5347.7: And the thing is that they aren't educated right now.
5348.1 --> 5350.5: If you don't own the keys, it's not your crypto.
5351.3 --> 5357.5: And of course, we should be pretty knowledgeable about that, mainly because we had Mt.
5357.5 --> 5359.7: Gox and we lost everything.
5361.0 --> 5362.6: History speaks for itself.
5362.9 --> 5364.5: But of course, people forget history.
5365.7 --> 5371.4: But with that said, I think it's also predicated with the fact of regulations and taxes.
5371.4 --> 5383.2: And now that they have this travel rule, it's more tedious to be able to, if you wanted to move your funds off, you know, off a centralized exchange to your own personal wallet.
5383.3 --> 5386.8: You need to stipulate what the reason is and whatnot.
5387.0 --> 5387.8: And this is fine.
5387.8 --> 5391.8: I think it's a good thing because the Japan government wants to know where is what.
5392.0 --> 5394.8: And are you the true owner or the beneficiary of this thing?
5395.0 --> 5396.6: And hence, they want to track it.
5396.7 --> 5397.8: And that's fine.
5398.5 --> 5410.6: But the thing is, is that I think it's also it's financial, too, mainly because if you move Bitcoin from one wallet to another wallet, there could be some sort of tax implications.

5411.2 --> 5416.4: And so people are hesitant or not educated well enough th
at maybe I just keep it there.
5416.7 --> 5422.0: And, you know, Bitcoin can move tomorrow at $125K versus
where it's at right now.
5422.6 --> 5426.4: And so I want to get that opportunity to have a liquidity
 option on the fly.
5426.4 --> 5438.2: And, of course, everything like this is all changing due
to, you know, everything's kind of working off centralized exchanges, working
 on DEXs and more liquidities on now DEXs than SEXs.
5438.2 --> 5451.5: So I think all in all, I do believe the public or at leas
t people like us are understanding that it's ever evolving.
5451.5 --> 5469.4: The landscape is changing and we need to get the mass pub
lic consumers that are into crypto and be able to understand that, yes, havin
g some on exchanges is probably a good idea maybe.
5469.4 --> 5484.8: But, you know, having if you have more than a certain amo
unt, you should probably take it off the exchange and have personal custody t
o that and have the educational materials around that and how to have that be
st practices like Eric has it in his safe.
5485.1 --> 5489.2: I have mine in my safe and I make sure that my seeds are
all separated.
5489.4 --> 5491.9: But I think I'm on an extreme case for that matter.
5492.8 --> 5498.3: So, yeah, that's my thoughts on kind of how where Japan i
s with it.
5498.3 --> 5513.9: But I think, you know, it's still I believe the majority
still like is like that in the United States, where in the US they probably h
ave it in some software based wallet on their mobile phone.
5514.3 --> 5527.7: And I've heard many, many stories where some people are c
oerced and, you know, forced to move funds off their even Coinbase account or
 from their software based wallets on their phones.
5527.7 --> 5532.3: And so I've heard many stories of different things that h
appened.
5532.7 --> 5537.6: And so it's like it's like perpetual trading.
5537.9 --> 5542.7: Do not put as much stuff that you're willing to lose in p
ublic's eye.
5543.5 --> 5547.7: It's best to just don't participate in purpose if you don
't want to lose the money.
5547.7 --> 5563.5: If you don't want to if you can if you don't mind losing
that money either on exchange or or on a software wallet on your mobile phone,
 then just don't keep it available and have it offline and just make it funda
mentally simple in that sense.
5568.4 --> 5569.4: Thank you.
5571.5 --> 5577.7: I'm from a word vendor used by Japanese customer crypto a
sset service provider used to bar.
5578.7 --> 5582.7: Our company name is ginkgo used by DM and Bitcoin.
5584.5 --> 5586.0: Please excuse my English.
5587.0 --> 5602.4: Listening to the discussion, we strongly recognized the i
mportance of researching SD and PP and security target and protection profile,
 especially since many enterprise wallets today is essentially a block block
boxes.
5603.2 --> 5611.4: We agree that resolving the black box issue is critical f

or future industry adoption.
5611.4 --> 5616.4: I have a pure exploratory question.
5617.3 --> 5621.1: Nothing is promised or guaranteed on our end.
5621.5 --> 5637.3: Of course, if enterprise world vendor like us who are abl
e to provide some level of information or operational insights would be valua
ble to this working group.
5638.3 --> 5648.4: And if so, what kind of specific cooperation or informati
on would be most helpful for developing the next step we just discussed?
5658.3 --> 5660.2: So thank you.
5660.3 --> 5666.2: Thank you for the question and suggesting sharing some in
formation.
5666.2 --> 5678.3: I think we moved a bit away from enterprise applications,
 but still for enterprise application, it could be interesting to know.
5679.8 --> 5692.3: It's not necessarily about the details of the of the wall
et, but more the level of assurance that you claim to be to be compliant or a
t least to be able to achieve.
5693.4 --> 5708.8: So level of assurance, which will help to understand basi
cally what are the requirements from the your customers, system integrators a
nd so on or exchange directly.
5710.3 --> 5717.2: That can be helpful in order to kind of frame what would
be the target for enterprise versus consumer, for instance.
5718.6 --> 5719.6: Thank you.
5721.2 --> 5725.8: We are, I think we are at the end of the session, so we n
eed to wrap up.
5726.6 --> 5729.2: We have not really defined what we are going to do.
5729.2 --> 5758.5: But I think at least trying to kind of describe different
 use cases and different expected level of assurance of some of the component
s for those use cases could make sense in terms of something that is achievab
le without too much effort as a first step.
5761.9 --> 5774.0: Well, I think the idea of some reference design is good,
but we need to be sure that to decide which use case we are looking at, becau
se otherwise we are going to address too many.
5778.6 --> 5798.8: And maybe as a takeaway, we can all think about for some
of the use cases, what should be really the lowest level of requirements in t
erms of functions and level of robustness, level of assurance, so that we at
some point we define something like this.
5798.8 --> 5803.1: I mean, we are not yet defining a theory, definitely not.
5803.2 --> 5807.2: Although the presentation has been interesting, of course,
 very useful.
5807.8 --> 5808.4: Thank you.
5809.5 --> 5816.5: But to be able to define a specific theory and the securi
ty target, we need to be sure which one we are looking at, of course.
5818.2 --> 5820.3: But still, thank you very much.
5820.7 --> 5823.1: That was very, very useful.
5823.7 --> 5832.3: We got quite a number of insights and food for thought fo
r the next few steps on this topic.
5832.6 --> 5833.2: Thank you, everyone.
5833.4 --> 5838.1: And I think we will start again in five, well, let's say
10 minutes.
5838.2 --> 5839.5: I don't know if there is a coffee today.

5839.6 --> 5840.0: 10 minutes.
5840.5 --> 5840.8: OK.
5841.4 --> 5843.5: So in 10 minutes, we start the next session.
5843.7 --> 5844.0: Thank you.
5883.0 --> 5883.1: Thank you.
5911.7 --> 5913.1: Thank you.
5941.7 --> 5943.1: Thank you.
5971.7 --> 5973.1: Thank you.
6001.7 --> 6003.1: Thank you.
6008.9 --> 6010.3: Thank you.