



Inhalt

1. Einführung	2
1.1 Hintergrund und Motivation	2
1.2 Ziele und Bedeutung	3
2. Distributed Ledger Technology (DLT)	4
2.1 Gründe für IOTA: Der Tangle im Vergleich zur Blockchain	5
2.2 IOTA – einige wichtige technische Grundlagen	6
2.3 Kryptografie	7
3. Grundlegende Architektur der Lösung	8
3.1 Schritt 1: Aufbau einer neuen „Collaboration“ zwischen User-A und User-B	9
3.2 Schritt 2: Übermittlung einer verschlüsselten Datei von User-A an User-B	11
3.3 Schritt 3: Entschlüsselung der empfangenen verschlüsselten Datei durch User-B	14
3.4 Schritt 4: Überprüfung der Unveränderlichkeit der geteilten Datei	17
4. Experimentelle Umsetzung und Validierung	19
4.1 Erstellung eines IOTA-Kontos	19
4.2 Erstellung eines kryptografischen Schlüssels für symmetrische Verschlüsselung	19
4.3 Erstellung eines öffentlichen und privaten Schlüsselpaares für asymmetrische Kryptografie ...	20
4.4 Erstellung von Dateien mit Level-1- und Level-2-Verschlüsselung	20
4.5 Erstellung von Hashes der Original- und verschlüsselten Dateien	20
4.6 IOTA-Transaktionen mit Metadaten und Prüfung der Authentizität	20
4.7 Diskussion der experimentellen Umsetzung	21
5. Praktische Anwendungsergebnisse der iPact-Smartphone-App	22
6. Überblick verwendete Softwaretechnologie	31
7. Fazit und Ausblick	33

1. Einführung

1.1 Hintergrund und Motivation

Personen und Organisationen verlassen sich beim Tausch von sensiblen Informationen zunehmend auf digitale Plattformen, Cloud-Speicherlösungen und Online-Kommunikationskanäle, um auch sensible Daten miteinander zu teilen. Diese zunehmende Digitalisierung bringt aber nicht nur Chancen und Prozesseffizienz mit sich, sondern auch neue Herausforderungen.

Ein zentraler Aspekt, neben der Sicherheit des Datenaustauschs, ist auch die Dokumentation der Tatsache, dass ein Austausch sensibler Daten stattgefunden hat. Wenn beispielsweise Universitäten mit anderen Forschungseinrichtungen oder mit Unternehmen an Forschungsprojekten arbeiten, müssen in der Regel Geheimhaltungsvereinbarungen unterzeichnet werden. Informationen, die unter diese Vereinbarungen fallen, werden heute überwiegend elektronisch ausgetauscht. Dabei stellt sich die Frage, wie dokumentiert werden kann, dass der Austausch tatsächlich stattgefunden hat, um später den Nachweis erbringen zu können, dass eine bestimmte Datei übertragen wurde und unverändert blieb. Oftmals fehlt bei aktuellen Methoden wie E-Mails, Cloud-Speichern oder Messenger-Diensten eine verlässliche Grundlage für die Beweisführung. Selbst im Falle eines E-Mail-Datenaustauschs hat der Sender zwar möglicherweise ein Sendeprotokoll, aber es bleibt unklar, ob der Empfänger die Nachricht tatsächlich erhalten hat. Darüber hinaus kann nicht automatisch eindeutig nachgewiesen werden, dass es sich bei der empfangenen Datei um dieselbe handelt, die ursprünglich gesendet wurde; diese Fragen betreffen zentrale Aspekte von Beweisbarkeit und Prozesssicherheit.

Zusätzlich sind Mechanismen erforderlich, die sicherstellen, dass der Austausch selbst auf sichere Weise erfolgt ist und dass eine Datei nach der Übertragung nicht unbefugt verändert wurde. Diese Herausforderungen sind in Summe nicht zufriedenstellend und zugleich kostengünstig und benutzerfreundlich gelöst.

Im Rahmen eines Teilprojektes des von der Stiftung für Innovation in der Hochschullehre geförderten InduKo-Projekts (Innovation durch Kollaboration) entstand iPact, eine Lösung zur Bewältigung der oben beschriebenen Herausforderungen. iPact ermöglicht es, den Austausch von Dateien sicher, einfach und ohne eine dritte Instanz zu dokumentieren, der vertraut werden müsste. Jede Transaktion wird in einem unveränderlichen Ledger aufgezeichnet, wodurch die Integrität der Datei und der gesamte Übertragungsprozess nachweisbar werden. iPact nutzt die Distributed Ledger Technology (DLT), um Dokumentation und Beweisführung ohne zentrale Infrastruktur zu gewährleisten.

1.2 Ziele und Bedeutung

iPact wurde entwickelt, um die oben beschriebenen zentralen Probleme des (manipulations-) sicheren und nachvollziehbaren Datenaustauschs, die insbesondere auch für forschende Organisationen und Personen eine Herausforderung sind, möglichst benutzerfreundlich zu lösen. Die Hauptziele des Projekts umfassen:

1. **Entwicklung eines sicheren, dezentralen Frameworks für den Dateiaustausch:**
iPact eliminiert die Notwendigkeit einer zentralen Kontrollinstanz durch die Nutzung von DLT. Benutzer behalten die volle Kontrolle über ihre Daten.
2. **Schaffung von digitalem Vertrauen durch Unveränderlichkeit:**
Durch die Nutzung der unveränderlichen Natur des DLT-Ledgers wird sichergestellt, dass alle Datenaustausche transparent aufgezeichnet und nicht manipuliert werden können.
3. **Förderung der Benutzerfreundlichkeit:**
Die Plattform wird speziell auch für nicht technische Nutzer entwickelt, um eine intuitive und leicht zugängliche Lösung ohne technische Hürden bereitzustellen.
4. **Realisierung einer rein DLT-basierten Lösung:**
iPact benötigt keine zusätzliche Infrastruktur wie ein Backend. Die gesamte Funktionalität basiert vollständig auf der IOTA-Technologie.
5. **Spätere Nachweisbarkeit auch ohne iPact:**
Der spätere Nachweis des Datenaustauschs und der Unveränderlichkeit von mit Hilfe von iPact getauschten Dateien ist rein mit den Standardwerkzeugen von IOTA möglich, auch selbst dann, wenn iPact später als Anwendung nicht mehr zur Verfügung stehen sollte.

Gerade die Bedeutung der letzten beiden Punkte 4. und 5. wurde uns erst im Laufe des Entwicklungsprozesses besonders bewusst und hat uns in der Umsetzung herausgefordert. Die Erfüllung dieser beiden Punkte bedeutet die Umsetzung von iPact als eine moderne, komplett dezentrale Applikation und eine Dokumentation der Ergebnisse in einer Form, die eine spätere Abfrage auch ohne iPact ermöglicht. Für uns waren diese Ziele sehr wichtig, weil ihre Erfüllung u.E. Vertrauen bei künftigen Nutzern in die Applikation schaffen.

iPact soll neben der konkreten Anwendung auch aufzeigen, wie dezentrale Technologien zukunftsweisende Lösungen schaffen können. Im Gegensatz zu herkömmlichen zentralisierten Ansätzen beseitigt iPact Schwachstellen wie den „Single Point of Failure“ und minimiert die Risiken durch zentrale Datenhaltung. Die Verteilung der Daten über das ein DLT erhöht die Sicherheit und Resilienz erheblich, sodass auch in unerwarteten Szenarien wie Systemausfällen oder Cyberangriffen ein hohes Maß an Schutz gewährleistet bleibt. Als manipulationssichere Lösung dient iPact somit auch als Anschauungsbeispiel für die Entwicklung weiterer Anwendungen, die auf Transparenz, Sicherheit und Dezentralisierung setzen. Die Einsatzmöglichkeiten von iPact akademischen Kontext sind sehr Interessent, gehen dabei aber über diesen hinaus. Neben dem Austausch vertraulicher Daten im Rahmen von Geheimhaltungsvereinbarungen eignet sich die Plattform auch für andere Szenarien, in

denen die Dokumentation des Austauschs und die Unveränderbarkeit von Dateien von zentraler Bedeutung sind – innerhalb und außerhalb der akademischen Welt.

2. Distributed Ledger Technology (DLT)

Um die Funktionsweise von iPact einordnen zu können, ist ein zumindest sehr grobes Basisverständnis für die zugrundeliegenden technischen Konzepte erforderlich. Die Distributed Ledger Technology (DLT) basiert auf einer dezentralen Netzwerkarchitektur und unterscheidet sich grundlegend von traditionellen, zentralisierten Systemen. Bei zentralisierten Systemen übernimmt eine zentrale Instanz, bspw. eine Bank, die Speicherung, Verwaltung und Validierung von Daten. DLT hingegen verteilt diese Aufgaben auf ein Netzwerk unabhängiger Teilnehmer bzw. Knoten.

DLT ist durch Dezentralität, Transparenz und Unveränderlichkeit gekennzeichnet. Die Dezentralität sorgt dafür, dass kein einzelner Knoten oder eine zentrale Instanz die Kontrolle über das gesamte Netzwerk bekommt. Transparenz wird dadurch gewährleistet, dass alle Knoten Zugriff auf die selbe Version des Ledgers haben. Dadurch können alle Teilnehmer des Netzwerks Transaktionen unabhängig überprüfen. Die Unveränderlichkeit des Ledgers wird durch kryptografische Mechanismen garantiert, die sicherstellen, dass einmal gespeicherte Daten nicht mehr geändert oder gelöscht werden können; das ist besonders wichtig, um Vertrauen zwischen den Teilnehmern zu schaffen und die Integrität der Daten zu gewährleisten. DLT ist ideal für Anwendungen, bei denen Sicherheit, Verlässlichkeit sowie Transparenz im Vordergrund stehen und bei der die Teilnehmer diese Ziele unabhängig von einer zentralen Instanz erreichen wollen.

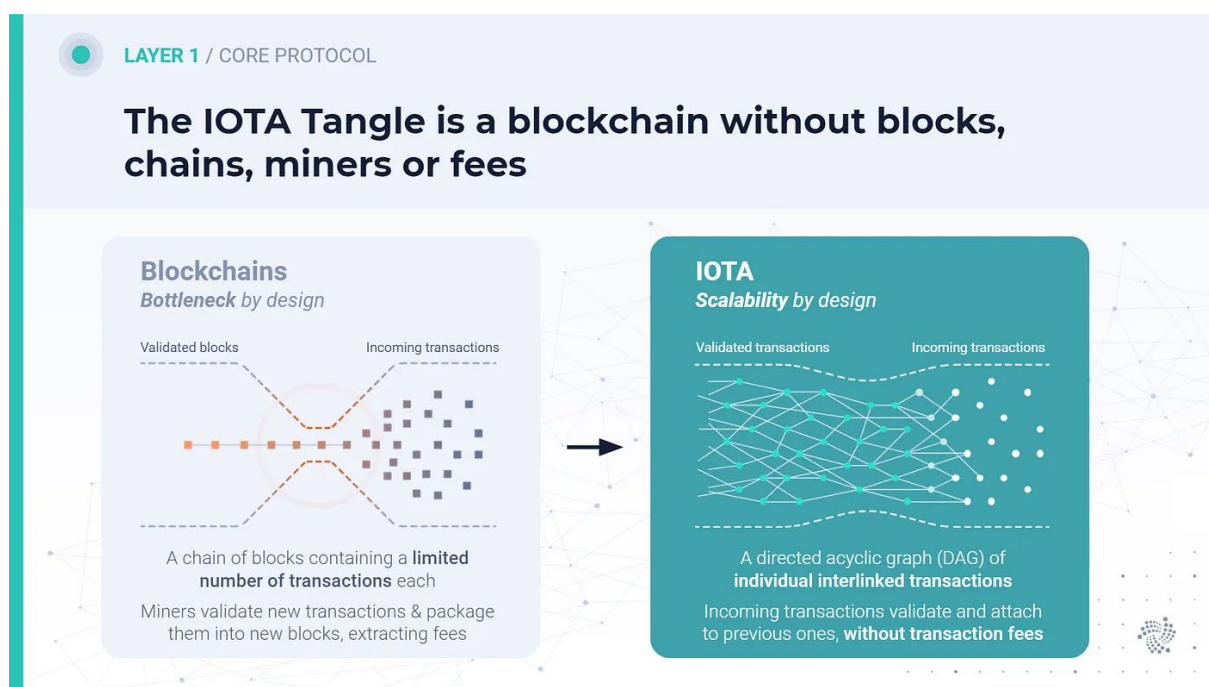
Innerhalb der DLT gibt es verschiedene Implementierungsansätze, wobei Blockchain und Directed Acyclic Graph (DAG) derzeit sehr relevante Umsetzungsformen sind. Blockchain, die traditionelle Form von DLT die insbesondere durch Bitcoin bekannt ist, besteht aus einer linearen Kette von Blöcken, die chronologisch und kryptografisch miteinander verknüpft sind. Jeder neue Block enthält Transaktionen sowie einen Hash des vorherigen Blocks, wodurch eine unveränderliche Kette entsteht. Konsensmechanismen wie Proof of Work (PoW) oder Proof of Stake (PoS) stellen sicher, dass alle Transaktionen validiert werden, bevor sie zur Blockchain hinzugefügt werden. Diese Mechanismen bieten hohe Sicherheit, sind jedoch mitunter sehr ressourcenintensiv und ihre Skalierbarkeit ist begrenzt. Die lineare Struktur der Blockchain bedeutet, dass Transaktionen sequenziell verarbeitet werden, was die Anzahl der Transaktionen pro Sekunde (TPS) einschränkt.

Im Gegensatz verwendet der Directed Acyclic Graph (DAG) keine lineare Struktur, sondern eine netzartige Architektur. DAG-basierte Systeme eignen sich besonders für Anwendungen, bei denen hohe Transaktionsvolumen, geringe Latenzzeiten und minimale Transaktionskosten sowie ein sehr geringer Ressourcenverbrauch angestrebt werden.

2.1 Gründe für IOTA: Der Tangle im Vergleich zur Blockchain

Die Wahl von IOTA als Basistechnologie für unser Projekt basiert insbesondere auf den spezifischen Eigenschaften des IOTA Tangle, die es gegenüber traditionellen Blockchain-Systemen vorteilhafter machen. Während Blockchain auf einer linearen Abfolge von Blöcken basiert, die durch Proof-of-Work (PoW) oder Proof-of-Stake (PoS) validiert werden, nutzt der IOTA Tangle eine Directed Acyclic Graph (DAG)-Struktur. Der IOTA Tangle war die erste Implementierung der DAG-Technologie und wurde mit dem Ziel entwickelt, auch die Anforderungen moderner Anwendungen im Bereich des Internets der Dinge (IoT) zu erfüllen.

Die untenstehende Abbildung illustriert die grundlegenden strukturellen Unterschiede zwischen traditionellen Blockchains und dem IOTA Tangle.



Quelle: <https://medium.com/@parecejacob/iota-bytes-assembly-of-data-the-new-eco-5f413953660>

Auf der linken Seite des Bildes wird das Konzept einer Blockchain dargestellt. Jeder Block enthält eine begrenzte Anzahl von Transaktionen und neue Transaktionen müssen von Minern validiert und in diese Blöcke integriert werden. Dieser Prozess führt zu einem inhärenten Engpass, da die Blockgröße und die Anzahl der Miner die Verarbeitungskapazität begrenzen. Zudem erfordern viele Blockchain-Systeme Gebühren für die Transaktionsverarbeitung, da Miner für ihre Arbeit incentiviert werden müssen, was bei IOTA nicht der Fall ist, warum hier auch keine Transaktionsgebühren anfallen.

Auf der rechten Seite zeigt die Abbildung die Funktionsweise des IOTA Tangles. Anstelle einer linearen Kette verwendet der Tangle eine gerichtete azyklische Graphenstruktur (DAG), eine parallelisierte Struktur. Transaktionen müssen nicht gesammelt werden, um dann linear miteinander verbunden zu werden, sondern sie können parallel an den verschiedenen Enden des DAG angefügt werden.

Der Proof-of-Work-Mechanismus, der z.B. bei Bitcoin für die Sicherheit des Netzwerks sorgt, verbraucht erhebliche Mengen an Energie, was nicht nur hohe Betriebskosten verursacht, sondern auch erhebliche Umweltfolgen hat. IOTA hingegen benötigt keine Miner und gilt als eine der ressourcenschonendsten DLTs überhaupt.

Von den zentralen Vorteilen Skalierbarkeit, Energieverbrauch und Transaktionskosten abgesehen, arbeitet die IOTA Foundation eng mit internationalen Organisationen, Regulierungsbehörden und gemeinnützigen Initiativen zusammen. Die IOTA Foundation, mit Standorten in Deutschland, der Schweiz und den Vereinigten Arabischen Emiraten, hat sich als vertrauenswürdiger Partner etabliert, insbesondere auch in Europa, wo sie an Projekten wie der European Blockchain Services Infrastructure (EBSI) beteiligt ist. Die European Blockchain Services Infrastructure ist eine Initiative u.a. der Europäischen Kommission mit dem Ziel eine europaweite Blockchain-Infrastruktur aufzubauen, die grenzüberschreitende öffentliche Dienstleistungen unterstützt. IOTA entspricht somit nicht nur den Anforderungen des europäischen Rechtsrahmens, sondern arbeitet auch aktiv an der Gestaltung von DLT-Standards mit.

Diese Eigenschaften machen IOTA zu einer technisch, ökologisch und ökonomisch nachhaltigen Wahl für Anwendungen, die auf eine sichere, effiziente und ressourcenschonende DLT-Infrastruktur angewiesen sind.

2.2 IOTA – einige wichtige technische Grundlagen

Das IOTA-Ökosystem umfasst mehrere Netzwerke, die verschiedene Anforderungen erfüllen. Das Mainnet dient als Produktionsumgebung für reale Transaktionen, während das Testnet als Sandbox für Entwickler genutzt wird, um neue Funktionen zu testen. Zusätzlich gibt es Shimmer, ein Staging-Netzwerk, das es Entwicklern ermöglicht, Protokolländerungen vor ihrer Einführung im Mainnet zu testen. Diese Netzwerke bieten eine flexible Infrastruktur, die es Entwicklern erleichtert, sichere und skalierbare Anwendungen wie iPact zu entwickeln und zu testen.

Jede Transaktion im IOTA Tangle besteht aus mehreren wesentlichen Komponenten. Die Inputs einer Transaktion repräsentieren ungenutzte Ausgaben aus vorherigen Transaktionen und definieren, welche Ressourcen für eine neue Transaktion verfügbar sind. Unlock Blocks enthalten digitale Signaturen, die bestätigen, dass die Inputs autorisiert sind. Diese Signaturen stellen sicher, dass nur der Eigentümer der Inputs Transaktionen initiieren kann, was unbefugte Zugriffe verhindert.

Outputs speichern die Ergebnisse der Transaktion und verbleiben im Tangle, bis sie weiterverwendet werden. Mechanismen wie die Expiration Unlock Condition legen Fristen fest, nach deren Ablauf ungenutzte Outputs an eine Rückgabeadresse übertragen werden. Die Timelock Unlock Condition verzögert den Zugriff auf Outputs bis zu einem bestimmten Zeitpunkt. Die Metadata-Funktion ermöglicht es, beliebige Daten innerhalb von Transaktionsausgaben zu speichern. Diese Daten bleiben dauerhaft im Netzwerk verfügbar und können auf allen Knoten abgerufen werden. Für iPact ist die Metadata-Funktion essenziell, um Informationen wie Dateibeschreibungen, Autorennennungen und Notizen direkt in der Transaktion zu verankern. Die Tag-Funktion erlaubt es, bis zu 64 Bytes indexierbarer Daten an eine Transaktionsausgabe anzuhängen. Diese Funktion ist besonders nützlich für iPact, da sie die Kategorisierung und Kennzeichnung von Transaktionen ermöglicht, beispielsweise mit Labels wie „Forschungszusammenarbeit“. Dadurch wird die Suche nach und Verwaltung von Datensätzen erheblich erleichtert. Die Tags werden, wie die Metadaten, als Binärdaten gespeichert und sind mit allen Knoten im Netzwerk kompatibel.

Schließlich sorgt die Storage Deposit Return Unlock Condition dafür, dass Benutzer eine Einlage für Speicherplatz hinterlegen müssen, die nach der Transaktion zurückerstattet wird. Dieser „Deposit“ stellt sicher, dass die Daten erhalten bleiben und nicht aus dem Ledger gelöscht werden. Insofern ist die Nutzung also, obwohl die Transaktion selbst keine Gebühren kostet, dennoch nicht ganz kostenlos, weil mit der nötigen Hinterlegung Kapital ohne Verzinsung gebunden wird. Insofern führen diese entgangenen Zinsen auf das gebundene Kapital doch zu gewissen, wenn auch für unseren Anwendungsfall absolut vernachlässigbar geringen, (Opportunitäts-)kosten.

Im IOTA-Protokoll werden sogenannte mnemonische Phrasen eingesetzt, um kryptografische Seeds auf eine benutzerfreundliche Weise zu generieren und zu verwalten. Eine mnemonische Phrase ist eine Folge von leicht verständlichen und aussprechbaren Wörtern, die Benutzern ermöglicht, komplexe kryptografische Schlüssel auf einfache Weise zu speichern und bei Bedarf wiederherzustellen. Diese Methode verbessert sowohl die Sicherheit als auch die Zugänglichkeit erheblich, da sie menschliche Fehler bei der Verwaltung kryptografischer Daten minimiert.

Eine mnemonische Phrase kann insofern mit einem Passwort verglichen werden, erfüllt jedoch eine spezifischere kryptografische Funktion. Sie besteht in der Regel aus 12, 18 oder 24 Wörtern, die aus einem festgelegten Wörterbuch ausgewählt werden, wie es beispielsweise der Standard BIP39 vorgibt. Jedes dieser Wörter repräsentiert einen Teil eines kryptografischen Schlüssels. Gemeinsam bilden sie den sogenannten Seed, der die Grundlage für die Generierung von privaten und öffentlichen Schlüsseln bzw. Adressen ist. Diese Wörter sind bewusst so gestaltet, dass sie leicht zu merken und sprachlich eindeutig sind. Eine Phrase wie „Apfel Fenster Katze Mond Baum usw.“ verbirgt einen komplexen kryptografischen Schlüssel, der den Zugriff auf ein Konto und die Durchführung von Transaktionen ermöglicht.

Ein Seed ist eine zufällige, kryptografisch generierte Zeichenfolge, die im IOTA-Netzwerk für alle wichtigen Operationen verwendet wird. Ein Seed fungiert als „Master-Schlüssel“, der den Zugriff auf alle mit dem Konto verbundenen Daten und Operationen ermöglicht. Wird ein Seed aus einer mnemonischen Phrase abgeleitet, kann er jederzeit reproduziert werden, solange die Phrase sicher aufbewahrt wurde. Dies bietet eine hohe Sicherheit und Wiederherstellbarkeit. iPact macht sich dieses Konzept zunutze, um die Verwaltung kryptografischer Schlüssel so einfach und sicher wie möglich zu gestalten. Im IOTA-Protokoll werden dann aus einem Seed öffentliche Adressen mithilfe von kryptografischen Algorithmen abgeleitet. Diese öffentlichen Adressen dienen als Identifikatoren, ähnlich wie Kontonummern in einem Bankensystem.

Der IOTA Explorer ist ein online Werkzeug, das es ermöglicht, alle Transaktionen und Daten, die im IOTA Tangle gespeichert sind, transparent einzusehen und zu überprüfen. Der Explorer unterstützt sowohl das Mainnet als auch das Testnet und ist somit vielseitig einsetzbar – von der Entwicklung neuer Anwendungen bis hin zur Überprüfung von Live-Daten. Für Anwendungen wie iPact bedeutet dies, dass alle im Tangle dokumentierten Vorgänge, wenn auch nicht komfortabel, jederzeit ohne zusätzliche Software überprüfbar wären.

2.3 Kryptografie

Kryptografie ist eine Technologie, um digitale Informationen vor unbefugtem Zugriff, Manipulation und Missbrauch zu schützen. Sie bietet die Grundlagen für sichere Kommunikation, Datenintegrität und die Authentifizierung von Nutzern und Systemen. Zwei zentrale Ansätze der Verschlüsselung sind die symmetrische und die asymmetrische Verschlüsselung, die jeweils auf spezifischen Prinzipien

beruhen. Ergänzend dazu spielt das Hashing eine entscheidende Rolle bei der Überprüfung der Integrität von Daten.

Symmetrische Verschlüsselung basiert darauf, dass derselbe Schlüssel sowohl zum Verschlüsseln als auch zum Entschlüsseln von Daten verwendet wird. Der Hauptnachteil ist hierbei normalerweise die Verteilung des Schlüssels: Beide Parteien müssen denselben Schlüssel sicher austauschen, was in vielen Szenarien eine Herausforderung ist.

Im Gegensatz dazu verwendet die asymmetrische Verschlüsselung ein Schlüsselpaar, bestehend aus einem öffentlichen Schlüssel (Public Key) und einem privaten Schlüssel (Private Key). Der öffentliche Schlüssel wird für die Verschlüsselung der Daten genutzt und kann sicher mit anderen geteilt werden, während der private Schlüssel ausschließlich dem Besitzer bzw. der Besitzerin bekannt ist und zur Entschlüsselung dient. Dies ermöglicht eine sichere Kommunikation ohne den direkten Austausch eines gemeinsamen Schlüssels.

Ein weiterer zentraler Aspekt der Kryptografie ist das sogenannte Hashing. Hierbei handelt es sich um einen Prozess, bei dem aus beliebigen Daten eine eindeutige, feste Länge an Zeichen (den sogenannten Hash) generiert wird. Hashing dient vor allem dazu, die Integrität von Daten zu überprüfen, da jede noch so kleine Änderung der Eingabedaten den resultierenden Hash vollständig verändert. Bekannte Algorithmen wie SHA-256 werden beispielsweise in Blockchain-Netzwerken verwendet, um Transaktionen zu sichern, oder auch in digitalen Signaturen, um die Authentizität von Dokumenten zu gewährleisten. Hashing ist deterministisch, das heißt, die gleiche Eingabe liefert immer denselben Hash. Zudem ist es praktisch, zumindest bis heute, unmöglich, aus einem Hash die ursprünglichen Daten zurückzugewinnen, was Hashing zu einem wichtigen Werkzeug für die Datensicherheit macht.

3. Grundlegende Architektur der Lösung

Die Entwicklung der grundlegenden Architektur und der finalen Lösung unseres Projekts war von Herausforderungen geprägt, die sowohl durch unsere eigenen, u.E. sehr wichtigen und sinnvollen Vorgaben, als auch durch technologische Veränderungen geprägt waren.

Während der Projektlaufzeit veröffentlichte IOTA ein umfangreiches Protokoll-Upgrade (Version 1.5, auch bekannt als „Chrysalis“), das die technologischen Grundlagen für unsere Entwicklung ganz erheblich veränderte. Dieses Update führte unter anderem das Shimmer-Testnets und inhaltlich neue Features ein, wodurch uns neue Möglichkeiten eröffnet wurden, aber auch neue Herausforderungen entstanden. So hat die technische Entwicklung der Grundlagentechnologie mit ihren weitreichenden Änderungen nicht nur denkbare Lösungen genommen, es wurden auch neue Möglichkeiten geschaffen; zu diesen neuen Funktionen zählte bspw. auch die Einführung von Smart Contracts. Wir haben die Nutzung von Smart-Contracts dann auch in unsere Überlegungen einbezogen und für spätere Entwicklungsstufen nicht ausgeschlossen, aber wir waren bestrebt, eine Lösung zu entwickeln die, wenn möglich, vollständig ohne zusätzliche Infrastruktur auf dem Layer-1, also sozusagen nativ innerhalb des IOTA-Kernprotokolls mit allen Vorteilen im Blick auf Kosten, Sicherheit und technische sowie ökonomische Nachhaltigkeit, umgesetzt werden kann.

Im Folgenden wird die grundlegende Architektur vorgestellt. Zunächst wird das sich vollständig auf IOTAs Layer-1 stützende Gesamtlösungskonzept beschrieben. Anschließend wird eine exemplarische technische Umsetzung dargestellt, die während des Projekts entwickelt wurde, um die Architektur zu

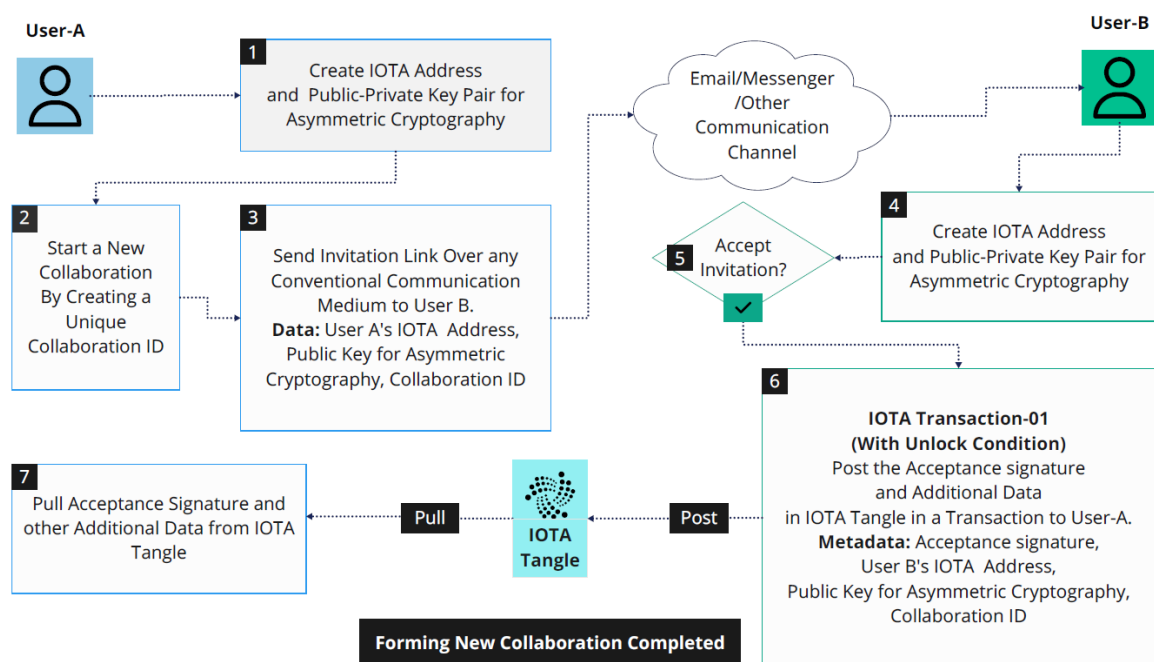
validieren. Im letzten Schritt wird dann die Implementierung als benutzerfreundliche Smartphone-Applikation beschrieben, die auf der Cross-Plattform-Entwicklungsumgebung Flutter basiert.

3.1 Schritt 1: Aufbau einer neuen „Collaboration“ zwischen User-A und User-B

Um den Prozess des Aufbaus einer „Collaboration“ bzw. Zusammenarbeit im Rahmen des iPact-Systems zu verstehen, ist es wichtig, zunächst zu klären, was unter einer „Collaboration“ bzw. "Zusammenarbeit" in diesem technischen Kontext zu verstehen ist. Eine Zusammenarbeit umfasst in diesem Fall die strukturierte Einrichtung einer Verbindung zwischen zwei Parteien, beispielsweise zwischen Instituten, Professoren, Forschungsgruppen, Unternehmen, die sensible Daten untereinander austauschen möchten. Dieser Austausch erfolgt häufig im Rahmen von Forschungsprojekten oder anderen Kooperationen, bei denen ein hohes Maß an Vertraulichkeit gewährleistet sein muss, etwa durch eine Geheimhaltungsvereinbarung (NDA).

Diese technische Abbildung einer Zusammenarbeit erfüllt mehrere wesentliche Funktionen: Sie dokumentiert technisch die Absicht und das gegenseitige Einverständnis der beteiligten Parteien, stellt sicher, dass die technologische Basis für den späteren sicheren Austausch von Daten geschaffen wird und bietet somit eine nachvollziehbare, strukturierte Grundlage. Die Erstellung dieser Struktur ist demgemäß die erste Phase einer sicheren Zusammenarbeit und legt den Grundstein für alle weiteren Schritte.

1 Forming New Collaboration Between User-A and User-B



Das Diagramm zeigt den Prozess der Erstellung einer „Collaboration“; im Folgenden werden die einzelnen Schritte erläutert:

1. **Erstellung der IOTA-Adresse und des Schlüsselpaares für User-A**

User-A beginnt den Prozess, indem er eine IOTA-Adresse erstellt, die als eindeutiger Identifikator dient und die Durchführung von Transaktionen auf dem Tangle ermöglicht. Gleichzeitig generiert User-A ein asymmetrisches Schlüsselpaar, bestehend aus einem öffentlichen und einem privaten Schlüssel. Diese Schlüssel werden für die spätere Ver- und Entschlüsselung im Rahmen des Datenaustausch benötigt.

2. **Erstellen einer einzigartigen Collaboration-ID**

Im nächsten Schritt erzeugt User-A eine eindeutige Collaboration-ID. Diese ID dient als unverwechselbarer Bezeichner für die Zusammenarbeit und wird benötigt, um die Kommunikation und die Interaktion zwischen den Teilnehmern eindeutig identifizieren und nachverfolgen zu können. Man kann sich diese eindeutige „Collaboration“ bildlich wie einen dedizierten Ordner vorstellen, den die Parteien künftig für den Austausch von Informationen verwenden werden.

3. **Versand der Einladung an User-B**

User-A verschickt die Einladung an User-B komfortabel über einen konventionellen Kommunikationskanal, beispielsweise per E-Mail oder Messenger. Die Einladung enthält die IOTA-Adresse von User-A, den öffentlichen Schlüssel für die asymmetrische Verschlüsselung und die Collaboration-ID. Diese Daten sind essenziell, damit User-B die Einladung überprüfen und bestätigen kann.

4. **Setup für User-B**

Nachdem User-B die Einladung erhalten hat, führt er ein vergleichbares Setup wie User-A durch. Er erstellt eine eigene IOTA-Adresse und generiert ein asymmetrisches Schlüsselpaar. Diese Schritte stellen sicher, dass User-B ebenfalls über die erforderlichen Ressourcen für die verschlüsselte Kommunikation und Authentifizierung verfügt.

5. **Annahme der Einladung:** User-B entscheidet, ob er die Einladung zur Zusammenarbeit annehmen möchte. Falls er sie ablehnt, wird der Prozess an dieser Stelle beendet. Wenn User-B die Einladung jedoch akzeptiert, folgt Schritt Nr. 6.

6. **Posten einer Transaktion auf dem IOTA-Tangle**

Nachdem User-B die Zusammenarbeit akzeptiert hat, erstellt er eine Transaktion auf dem IOTA-Tangle. Diese Transaktion enthält Metadaten, die die Zusammenarbeit dokumentieren. Zu den Metadaten gehören:

- Die Annahmesignatur, die die Zustimmung von User-B bestätigt.
- Die IOTA-Adresse und der öffentliche Schlüssel von User-B.
- Die Collaboration-ID, um die Transaktion eindeutig zuzuordnen.

- Zusätzlich hinterlegt User-B eine minimale Menge an IOTA-Tokens als Sicherheit, um die Daten vor zukünftiger Löschung zu schützen (vgl. Kapitel 2.2).
- Diese Tokens, und damit auch die mit dieser Transaktion verbundenen Metadaten, bleiben für eine langfristige Dauer von beispielsweise 50 Jahren (im Code steuerbar) gesperrt bzw. vor Löschung geschützt.

7. Bestätigung der Zusammenarbeit

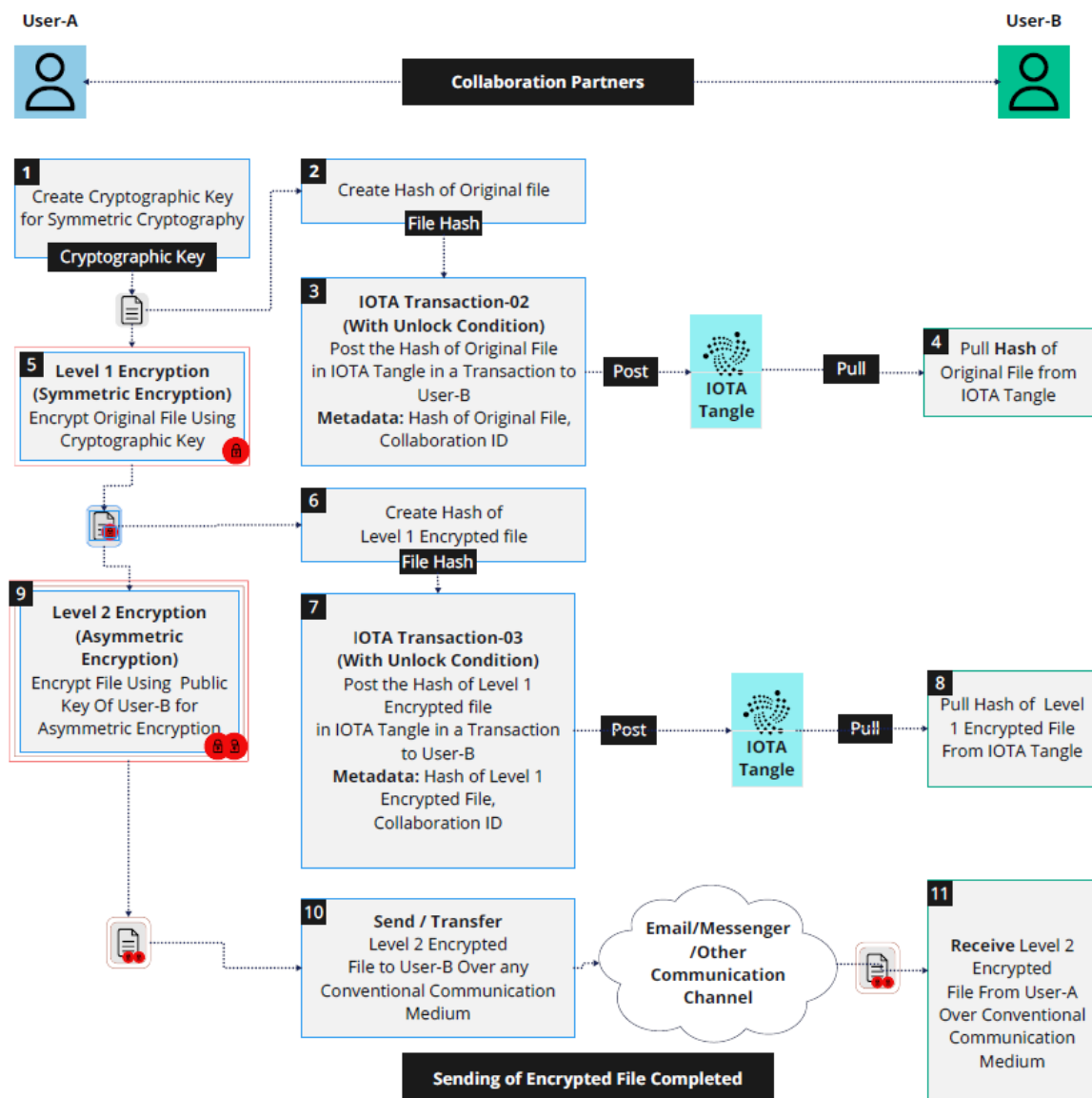
User-A ruft die Transaktionsdaten von User-B aus dem IOTA-Tangle ab. Diese Daten umfassen die Annahmesignatur, die IOTA-Adresse, den öffentlichen Schlüssel und die Collaboration-ID von User-B. Durch diesen Schritt wird die Zusammenarbeit offiziell abgeschlossen, und beide Teilnehmer können sicher miteinander interagieren.

Nach Beendigung dieses Prozesses wurde eine sichere und überprüfbare Zusammenarbeit zwischen User-A und User-B erstellt und auf dem DLT („Blockchain“) dokumentiert.

3.2 Schritt 2: Übermittlung einer verschlüsselten Datei von User-A an User-B

Die Übertragung einer Datei von User-A zu User-B geht weit über die reine Verschlüsselung und Übermittlung hinaus. Eine zentrale Herausforderung war es, eine manipulationssichere Methode zu entwickeln, die zweifelsfrei dokumentiert, dass der Empfänger die Datei nicht nur erhalten hat, sondern auch tatsächlich darauf zugreifen kann. Zudem musste gewährleistet sein, dass die empfangene Datei exakt mit der gesendeten übereinstimmt. Die Lösung dieses „Handshake-Mechanismus“ wurde letztlich durch eine Nutzung von auf dem IOTA Layer-1 verfügbaren Funktionen in Kombination mit bewährten kryptografischen Verfahren und einem entsprechenden Prozessdesign möglich. Im Folgenden werden die Schritte im Detail erläutert.

2 Sending Encrypted File From User-A to User-B



1. Erstellung eines symmetrischen Schlüssels durch User-A

User-A generiert einen kryptografischen Schlüssel für symmetrische Verschlüsselung. Dieser Schlüssel bildet die Grundlage für die erste Verschlüsselung der Datei, also für die „innere Hülle“ der Verschlüsselung.

2. Erzeugung eines Hashwerts der Originaldatei

User-A erstellt mit einem Hashing-Algorithmus (z. B. SHA-256) einen eindeutigen Hashwert der Originaldatei. Dieser Hash dient als unveränderlicher Fingerabdruck der Datei und wird später genutzt, um sicherzustellen, dass die Datei während des Prozesses nicht manipuliert wurde.

3. Posten des Hashwerts der Originaldatei im IOTA Tangle

Der Hashwert der Originaldatei wird zusammen mit der Collaboration-ID als Transaktion auf dem IOTA Tangle veröffentlicht. Diese Transaktion wird mit einer zeitlichen Sperre (Lock Condition) versehen, sodass der Tokenbetrag der Transaktion für eine festgelegte Dauer (z. B. 50 Jahre) gesichert bleibt. Diese Sperre schützt die Transaktionsinformationen vor unbefugtem Zugriff und sichert ihre langfristige Verfügbarkeit.

4. Abrufen des Hashwerts durch User-B

User-B lädt den im Tangle gespeicherten Hash der Originaldatei herunter. Dieser Hash ermöglicht es User-B später, die Integrität der empfangenen Datei zu überprüfen.

5. Erste Verschlüsselungsebene (symmetrisch)

User-A verschlüsselt die Originaldatei mit dem zuvor generierten symmetrischen Schlüssel. Diese Verschlüsselungsebene schützt die Datei zwar auch vor unbefugtem Zugriff während der Übertragung, aber vor allen Dingen wird Gleichzeitig mit dieser Verschlüsselung eine wesentliche Voraussetzung für den Handshake-Mechanismus implementiert: Die spätere Entschlüsselung dieser Ebene durch User-B beweist, dass die Datei empfangen wurde und dass Zugriff durch User B möglich ist.

6. Erzeugung eines Hashwerts der Level-1-verschlüsselten Datei

User-A erzeugt einen neuen Hashwert aus der Level-1-verschlüsselten Datei. Dieser Hash wird erneut im Tangle veröffentlicht, um sicherzustellen, dass die verschlüsselte Datei nach der erstmaligen Verschlüsselung unverändert bleibt.

7. Posten des Hashwerts der Level-1-verschlüsselten Datei im IOTA Tangle

Der Hashwert der Level-1-verschlüsselten Datei wird zusammen mit der Collaboration-ID als zweite Transaktion auf dem Tangle gespeichert. Diese Transaktion ist ebenfalls durch eine zeitliche Sperre geschützt, was ihre Unveränderlichkeit gewährleistet.

8. Abrufen des Hashwerts der Level-1-verschlüsselten Datei durch User-B

User-B lädt den Hashwert der Level-1-verschlüsselten Datei herunter. Dieser Schritt ermöglicht eine zusätzliche Verifizierungsebene, die sicherstellt, dass die verschlüsselte Datei während der Übertragung nicht manipuliert wurde.

9. Zweite Verschlüsselungsebene (asymmetrisch)

User-A verschlüsselt die bereits symmetrisch verschlüsselte Datei (Level-1-verschlüsselte Datei) mit dem öffentlichen Schlüssel von User-B. Diese zweite Verschlüsselungsebene gewährleistet, dass nur User-B, der über den zugehörigen privaten Schlüssel verfügt, die Datei entschlüsseln kann.

10. Übertragung der Level-2-verschlüsselten Datei

Die zweifach verschlüsselte Datei wird über einen herkömmlichen Kommunikationskanal (z. B. E-Mail, Messenger) an User-B gesendet. Diese Methode ist sicher, selbst wenn der Kommunikationskanal kompromittiert wäre, weil der Zugriff auf die Datei nur mit den beiden erforderlichen Schlüsseln möglich ist.

11. Empfang durch User-B

User-B empfängt die zweifach verschlüsselte Datei, die Datei ist bereit für die Entschlüsselung. Dieser Prozess und die zugehörige Dokumentation wird im Folgenden beschrieben.

3.3 Schritt 3: Entschlüsselung der empfangenen verschlüsselten Datei durch User-B

Nach dem Erhalt der zweifach verschlüsselten Datei beginnt User-B mit der Entschlüsselung, um den ursprünglichen Dateiinhalt zugänglich zu machen. Diese Phase, wie in Abbildung 3 dargestellt, beschreibt die einzelnen Schritte zur sicheren Entschlüsselung, Verifizierung und Bestätigung der Datenintegrität. Ziel ist es, sicherzustellen, dass die Datei unverändert und ausschließlich für User-B zugänglich bleibt.

1. Erste Entschlüsselung (Asymmetrische Entschlüsselung)

User-B entschlüsselt die zweifach verschlüsselte Datei mithilfe seines privaten Schlüssels. Diese Schicht der Entschlüsselung entfernt die äußere „Hülle“, die User-A zuvor mit dem öffentlichen Schlüssel von User-B verschlüsselt hat. Dadurch wird die Level-1-verschlüsselte Datei freigelegt.

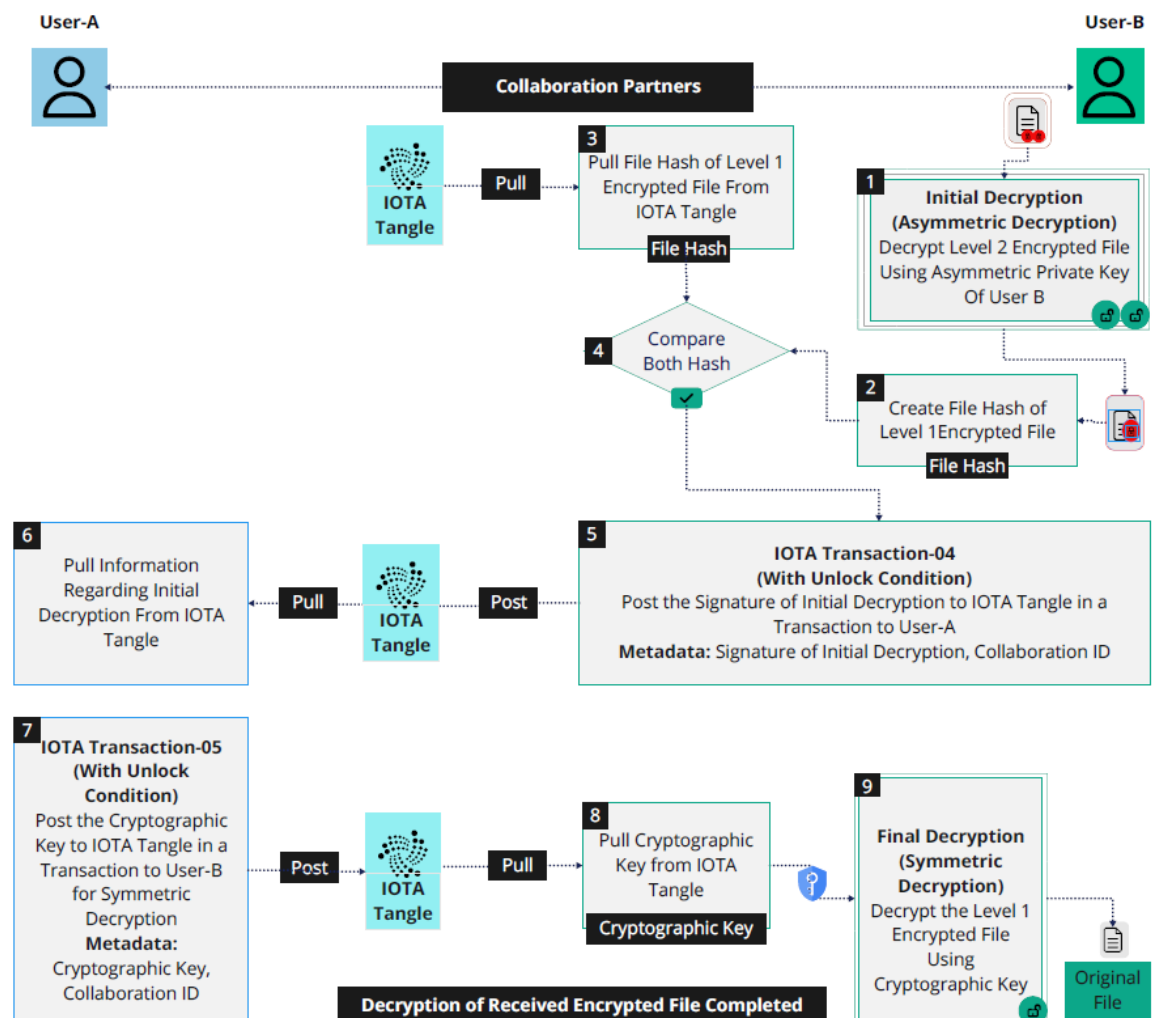
2. Erstellung des Hashs der Level-1-Datei

Nachdem die Level-1-verschlüsselte Datei entschlüsselt wurde, erzeugt User-B mithilfe des standardisierten Hashing-Algorithmus, der auch von User-A verwendet wurde, einen Hash dieser Datei. Dieser Hash, der als Fingerabdruck der Level-1 Datei fungiert, kann später zur Verifizierung der Datenintegrität verwendet.

3. Abrufen des Level-1-Dateihashes vom IOTA Tangle

User-B greift auf den zuvor von User-A veröffentlichten Hash der Level-1-verschlüsselten Datei zu, der als unveränderlicher Referenzpunkt auf der IOTA Tangle gespeichert wurde. Dieser Hash wurde während der Phase des Dateitransfers von User-A hochgeladen, um die Integrität der Datei zu garantieren.

3 Decryption of received encrypted file by User-B



4. Verifizierung der Datei-Integrität

User-B vergleicht den lokal generierten Hash der Level-1-Datei mit dem Hash, der vom IOTA Tangle abgerufen wurde. Stimmen beide Hashes überein, bestätigt dies, dass die Level-1-Datei während des Transfers weder verändert noch manipuliert wurde.

5. Bestätigung der Entschlüsselung

Nach erfolgreicher Verifizierung der Datei-Integrität veröffentlicht User-B eine Transaktion auf dem IOTA Tangle. Diese Transaktion enthält eine Signatur, die die erfolgreiche Entschlüsselung der Datei bis zur ersten Ebene dokumentiert. Zudem enthält die Transaktion die Kollaborations-ID und relevante Metadaten. Damit wird User-A darüber informiert, dass User-B die Datei erfolgreich entschlüsselt und verifiziert hat – User B besitzt die Datei also und hat Zugriff, die Datei ist nicht korumpiert.

6. Verifizierung der Entschlüsselungsbestätigung

User-A ruft die Bestätigungsdaten aus der entsprechenden Transaktion auf der IOTA Tangle ab. Dies dient als Nachweis dafür, dass User-B die Datei korrekt empfangen, entschlüsselt und die Integrität verifiziert hat. Für User-A ist das ein entscheidender Schritt, um sicherzustellen, dass der Prozess wie vorgesehen abgelaufen ist und alle Voraussetzung für den Abschluss des Prozesses vorliegen.

7. Veröffentlichung des symmetrischen Schlüssels

Nach der Verifizierung durch User-A wird der für die Level-1-Verschlüsselung verwendete symmetrische Schlüssel von User-A auf dem IOTA Tangle gespeichert. Der Schlüssel wird als zeitgesperrte Transaktion veröffentlicht, was bedeutet, dass User-B innerhalb eines definierten Zeitraums darauf zugreifen kann.

8. Abrufen des symmetrischen Schlüssels

User-B ruft den symmetrischen Schlüssel aus der entsprechenden Transaktion auf dem IOTA Tangle ab. Dieser Schlüssel ist erforderlich, um die letzte Verschlüsselungsschicht zu entfernen und Zugriff auf die ursprüngliche Datei zu erhalten.

9. Endgültige Entschlüsselung

Mit dem symmetrischen Schlüssel entschlüsselt User-B die Level-1-verschlüsselte Datei. Dieser letzte Schritt gibt die Originaldatei frei und markiert den Abschluss des Prozesses.

Der vorgestellte Prozess implementiert einen manipulationssicheren Handshake, der durch die Kombination aus kryptografischen Methoden und der unveränderlichen Dokumentation im IOTA Tangle gewährleistet wird. Wichtige Punkte des Sicherheitsmechanismus sind:

1. Unveränderliche Dokumentation:

Jeder Schritt – von der Hash-Erstellung bis zur Veröffentlichung des Schlüssels – wird auf dem Tangle dokumentiert. Dies bietet eine transparente und manipulationssichere Nachverfolgbarkeit.

2. Bestätigung des Zugriffs:

Die Entschlüsselung der Level-2-Verschlüsselung durch User-B und die anschließende Dokumentation im Tangle beweisen, dass User-B die Datei nicht nur erhalten, sondern auch erfolgreich darauf zugreifen konnte.

3. **Finale Schlüsselübergabe:**

Die Freigabe des symmetrischen Schlüssels durch User-A erfolgt erst nach der Bestätigung von User-B im Tangle, was die Sicherheit und Integrität des gesamten Prozesses gewährleistet.

4. **Langfristige Sicherheit:**

Durch die zeitliche Sperre der Transaktionen wird sichergestellt, dass die hinterlegten Daten langfristig zugänglich bleiben, ohne dass eine Manipulation oder Löschung möglich ist.

Durch die innovative Kombination aus asymmetrischer und symmetrischer Verschlüsselung mit der Dokumentation über den IOTA Tangle wird sichergestellt, dass der gesamte Austauschprozess in jeder Phase lückenlos nachvollziehbar bleibt. Dabei wird nicht nur dokumentiert, dass User-B die Datei erhalten hat, sondern auch zweifelsfrei nachgewiesen, dass er tatsächlich Zugriff darauf hatte. Erst nach dieser Bestätigung wird der finale symmetrische Schlüssel freigegeben, was eine manipulationsfreie Übergabe garantiert. Alle Schritte sind so gestaltet, dass kein Vertrauen in Dritte erforderlich ist, da die Sicherheit und Transparenz vollständig durch kryptografische Verfahren und die dezentrale Natur des IOTA Tangle gewährleistet werden.

Diese Lösung zeigt, dass selbst unter den gewählten Anforderungen – wie der ausschließlichen Nutzung von Layer-1-Funktionalitäten ohne externe Infrastruktur – ein Höchstmaß an Sicherheit, Transparenz und langfristiger Nachvollziehbarkeit erreicht werden kann.

3.4 Schritt 4: Überprüfung der Unveränderlichkeit der geteilten Datei

Nach Abschluss der Phasen des Datenaustauschs und der Entschlüsselung können sowohl User-A als auch User-B die Unveränderlichkeit der geteilten Datei unabhängig überprüfen. Diese Prüfung stellt sicher, dass die Datei über die Zeit hinweg unverändert geblieben ist, man kann also auch Jahre später noch nachweisen, um welche Datei es sich gehandelt, obwohl die Datei selbst nicht im IOTA Tangle gespeichert werden kann. Der Prozess, wie in Abbildung 4 dargestellt, nutzt kryptografisches Hashing und die unveränderlichen Aufzeichnungen auf dem IOTA Tangle, um sicherzustellen, dass keine Modifikationen an der Datei vorgenommen wurden, seit sie geteilt wurde.

1. **Erzeugung eines aktuellen Datei-Hashes**

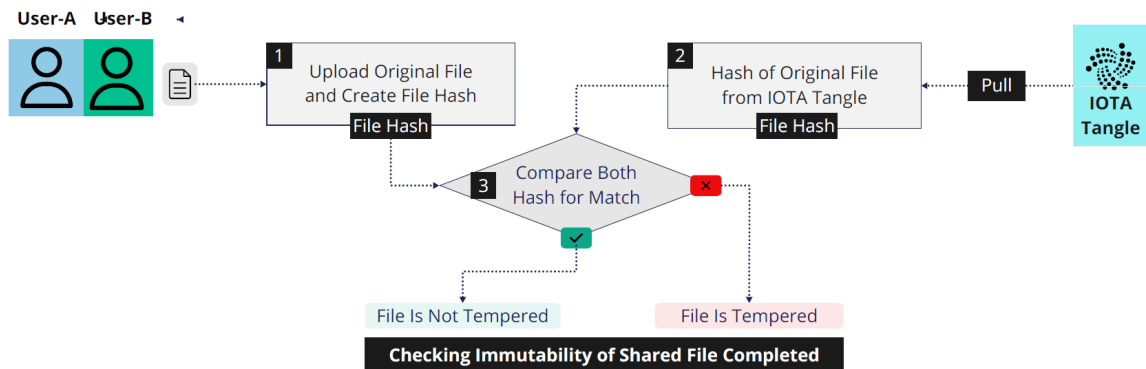
Der jeweilige Nutzer (entweder User-A oder User-B) lädt die betreffende Datei hoch und erzeugt deren aktuellen Hashwert mit dem gleichen kryptografischen Hash-Algorithmus, der während des ursprünglichen Dateiübertragungsprozesses verwendet wurde (z. B. SHA-256). Dieser Hashwert dient als aktueller „Fingerabdruck“ der Datei und repräsentiert ihren momentanen Zustand.

2. **Abruf des gespeicherten Datei-Hashes aus dem IOTA Tangle**

Der Nutzer ruft den ursprünglichen Datei-Hash aus dem IOTA Tangle ab, der während der initialen Dateiübertragung gespeichert wurde. Dieser gespeicherte Hashwert dient als überprüfbarer Referenz für den ursprünglichen Zustand der Datei und bildet die Grundlage für die Integritätsprüfung.

4 Check Immutability of Shared File

Is the File tempered or not?



3. Vergleich der Hashes zur Integritätsprüfung

Der aktuelle, neu generierte Hashwert der Datei wird mit dem auf dem IOTA Tangle gespeicherten ursprünglichen Hash verglichen. Stimmen die beiden Hashwerte überein, bestätigt dies, dass die Datei seit ihrer ersten Übertragung nicht verändert oder manipuliert wurde, und ihre Integrität bleibt intakt. Falls die Hashwerte jedoch nicht übereinstimmen, deutet dies auf Änderungen an der Datei hin, was bedeutet, dass die Unveränderlichkeitsprüfung nicht bestanden wurde.

4. Abschluss der Unveränderlichkeitsprüfung

Nach dem Vergleich der Hashwerte erhält der Nutzer eine klare Bestätigung darüber, ob die Datei weiterhin in ihrem ursprünglichen, unveränderten Zustand vorliegt. Dieser Verifizierungsprozess kann jederzeit wiederholt werden, wodurch die Kollaborationspartner die Authentizität der Datei vor allen Dingen auch zu einem späteren Zeitpunkt unabhängig überprüfen können.

Der in den vorangegangenen Kapiteln beschriebene Prozess beschreibt ein aufeinander abgestimmtes Zusammenspiel verschiedener Mechanismen, die darauf ausgelegt sind, den manipulationssicheren und transparenten Austausch sensibler Dateien zu gewährleisten – eine innovative Lösung, die allerdings auch eine entsprechende Komplexität mit sich bringt. Daher liegt die nächste Herausforderung nicht nur in der technischen Validierung des Konzepts, um dessen Funktionalität wie geplant zu bestätigen, sondern auch in der Entwicklung einer dennoch benutzerfreundlichen Smartphone-App. Diese soll es den Nutzern ermöglichen, den Austausch effizient zu organisieren – insbesondere auch dann, wenn es um mehrere Dateien geht, was ja in der Praxis häufig der Fall ist. Im folgenden Kapitel wird zunächst die technische Validierung des Konzepts knapp beschrieben, bevor anschließend die entwickelte Smartphone-App detailliert vorgestellt wird.

4. Experimentelle Umsetzung und Validierung

Zur Validierung der vorgeschlagenen Framework-Architektur wurden eine Reihe von Experimenten durchgeführt, um die Umsetzbarkeit des Frameworks zu bewerten. In diesem Kapitel werden die wesentlichen experimentellen Implementierungen beschrieben, die sich auf die Integration von IOTA und kryptografische Funktionen beziehen. Für IOTA-bezogene Operationen wurden das IOTA SDK und die IOTA Stronghold-Bibliothek in einer Rust-Umgebung verwendet. Für die kryptografischen Funktionen kamen die RSA- und Fernet-Bibliotheken in Python zum Einsatz.

4.1 Erstellung eines IOTA-Kontos

Die folgende Abbildung zeigt den Prozess der Erstellung eines IOTA-Kontos.

```
Generated mnemonic: "cube barely trash upon pumpkin arch food tackle maximum ugly isolate depart whale plate quit lamp swear pill unique fly company century casual oak"  
IOTA Public Address: AccountAddress { address: Bech32Address(rms1qqc3vfktzy90ken2dz9et30ejygy30gtp94fdd2jmvhd63pg9a7krjacpf), key_index: 0, internal: false, used: false }
```

Erstellung eines IOTA-Kontos

1. Generierung der mnemonischen Phrase

Eine zufällige mnemonische Phrase wird generiert, die als Grundlage zur Ableitung eines einzigartigen Seeds für das IOTA-Konto dient. Dieser Seed wird benötigt, um ein IOTA-Konto sicher zu erstellen oder wiederherzustellen.

2. Erstellung des Kontos

Mithilfe der mnemonischen Phrase werden eine Wallet und eine Stronghold-Datei erstellt. Die Stronghold-Datei fungiert als lokale „Tresordatenbank“, die eine sichere Synchronisation mit dem IOTA Tangle ermöglicht und die Wiederherstellung des Kontos bei Bedarf erlaubt. Während der Kontoerstellung wird eine Ed25519-Adresse generiert, die als IOTA-Wallet-Adresse dient und für Transaktionen auf dem IOTA Tangle erforderlich ist.

3. Verifizierung

Es wurden eindeutige IOTA-Konten und Adressen für User-A und User-B erstellt (IOTA Shimmer Testnet Explorer).

4.2 Erstellung eines kryptografischen Schlüssels für symmetrische Verschlüsselung

Für die Verschlüsselung auf Level 1 wird ein kryptografischer Schlüssel benötigt, der mit Hilfe der Python Fernet-Bibliothek erstellt wurde:

```
Generated Cryptographic Key For Symmetric Encryption: b'Xcv6DsnAvIyNgUe6bNZeR6yfApoyIumrU_aJZ4TyP2k='
```

Erstellung eines kryptografischen Schlüssels für symmetrische Verschlüsselung

1. Schlüsselerstellung

Ein symmetrischer Verschlüsselungsschlüssel wurde mithilfe der Fernet-Bibliothek generiert.

2. Sicherheitsaspekte

Der generierte Schlüssel wurde sicher gespeichert und nur entsprechend den Zugriffs- und Verifizierungsprotokollen des Frameworks weitergegeben.

4.3 Erstellung eines öffentlichen und privaten Schlüsselpaares für asymmetrische Kryptografie

Für die asymmetrische Kryptografie wurde der RSA-Algorithmus verwendet, ein weit verbreitetes Verfahren. Die folgende Abbildung zeigt die Schritte und Ergebnisse der Generierung eines öffentlichen und privaten Schlüsselpaares mit RSA.

```
RSA Public Key: PublicKey(1878051198618383598984522509492639909296882548562279955961108433719979538506915408449390528333160810400255047272262588644575)
RSA Private Key: PrivateKey(18780511986183835989845225094926399092968825485622799559611084337199795385069154084493905283331608104002550472722625886445)
```

Erstellung eines öffentlichen und privaten Schlüsselpaares für asymmetrische Kryptografie

1. **RSA-Schlüsselerstellung**

Mithilfe des RSA-Algorithmus wurde ein öffentliches und privates Schlüsselpaar erstellt, um einen sicheren Datenaustausch zwischen User-A und User-B zu ermöglichen.

2. **Schlüsselverteilung:**

Der öffentliche Schlüssel wurde mit den Kollaborationspartnern geteilt, um sichere verschlüsselte Kommunikation zu gewährleisten. Der private Schlüssel wurde von jedem Nutzer sicher verwahrt, um Dateien entschlüsseln zu können.

4.4 Erstellung von Dateien mit Level-1- und Level-2-Verschlüsselung

Der Datei-Verschlüsselungsprozess bestand aus zwei Stufen, um die Datensicherheit zu maximieren:

1. **Level-1-Verschlüsselung:**

Die Originaldatei wurde mithilfe des durch symmetrische Verschlüsselung (Fernet) generierten kryptografischen Schlüssels verschlüsselt.

2. **Level-2-Verschlüsselung:**

Die Level-1-verschlüsselte Datei wurde zusätzlich mit dem öffentlichen Schlüssel von User-B verschlüsselt, der während der RSA-Schlüsselpaarerstellung erzeugt wurde.

4.5 Erstellung von Hashes der Original- und verschlüsselten Dateien

Zur Verifizierung und Sicherstellung der Unveränderlichkeit wurde ein Hash sowohl für die Originaldatei als auch für die Level-1-verschlüsselte Datei generiert. Die Abbildung zeigt die Schritte und Ergebnisse der Hash-Erstellung:

```
sha256 Hash of Original File 6b53c96e1160082e838ae759a5d104ac9a977cccbea8d0c00df6925791938381
sha256 Hash of Level 1 Encrypted File 6b53c96e1160082e838ae759a5d104ac9a977cccbea8d0c00df6925791938381
```

Erstellung von Hashes der Original- und verschlüsselten Dateien

Mithilfe der FileHash-Bibliothek und des SHA-256-Hashing-Algorithmus wurden eindeutige Hashes für die Originaldatei und die Level-1-verschlüsselte Datei erstellt.

4.6 IOTA-Transaktionen mit Metadaten und Prüfung der Authentizität

Im Folgenden wurden fünf IOTA-Transaktionen durchgeführt, die jeweils Metadaten enthielten, um den Prozess wie zuvor beschrieben abbilden zu können. Während der initialen Phase der Kollaborationsetablierung postet User-B eine IOTA-Transaktion (Transaction-01) an User-A. Diese Transaktion enthält die IOTA-Adresse von User-B, die Kollaborationsakzeptanz-Signatur, den

öffentlichen Schlüssel und die Kollaborations-ID. Die Transaktion wurde für 50 Jahre gesperrt, um die Integrität und langfristige Verfügbarkeit der Transaktion sicherzustellen. Die folgende Abbildung zeigt die Details von Transaction-01 im IOTA Shimmer Testnet Explorer:

```
Collaboration CLB0123456 Creation Completed  
Transaction sent: 0x4437a3c11e6d5c0772657cf51d27e523c394271823458c00ae3c7c4b6a5f1c1d  
Block sent: https://explorer.shimmer.network/testnet/block/0xa446fd742bb966943890201d662c0bea5a93030e9c7b91c87f0f5226244da6c5
```

Transaktion der Kollaborationsakzeptanz-Signatur

Um sicherzustellen, dass die ausgetauschten Dateien jederzeit überprüfbar bleiben und ihre Integrität gewahrt wird, speichert User-A die Hashes der Originaldatei und der verschlüsselten Dateien nacheinander auf dem IOTA Tangle: User-A integriert den Hash der Originaldatei in Transaktion-02 und sendet diese an User-B. Der Hash der Level-1-verschlüsselten Datei wird in Transaktion-03 geteilt; beide Transaktionen wurden auf dem IOTA Tangle aufgezeichnet und mit zeitlichen Sperrbedingungen versehen.

Nach dem „Erhalt“ der Level-2-verschlüsselten Datei verwendet User-B seinen privaten Schlüssel, um die Level-2-verschlüsselte Datei zu entschlüsseln. Anschließend postet User-B Transaktion-04, um die erfolgreiche erste Entschlüsselung zu bestätigen.

Im Anschluss postet User-A Transaktion-05, die den kryptografischen Schlüssel enthält, der für die endgültige Entschlüsselung durch User-B erforderlich ist. Mit diesem Schlüssel entschlüsselt User-B die nun Level-1-verschlüsselte Datei und erhält die Originaldatei.

Das Test-Framework ermöglicht beiden Nutzern, die Authentizität der Datei zu überprüfen, indem der aktuelle Datei-Hash mit dem in Transaktion-02 gespeicherten Hash verglichen wird:

```
sha256 Hash of Original File obtained from Final Decryption: 6b53c96e1160882e838ae759a5d104ac9a977cccbea8d0c00df6925791938381  
sha256 Hash of Original File stored in IOTA Transaction-02: 6b53c96e1160882e838ae759a5d104ac9a977cccbea8d0c00df6925791938381
```

4.7 Diskussion der experimentellen Umsetzung

Die Experimente zur Validierung des vorgeschlagenen Frameworks haben gezeigt, dass das Konzept erfolgreich umgesetzt werden kann. Die Ergebnisse bestätigen, dass die Kombination aus dualer Verschlüsselung, kryptografischen Verfahren und der Nutzung der dezentralen Struktur des IOTA-Tangles eine sichere und manipulationsresistente Grundlage für den Austausch sensibler Dateien schafft.

Die dezentrale Architektur des Frameworks ermöglicht es wie angestrebt, ohne zentrale Server zu operieren und dennoch ein hohes Maß an Vertrauen zwischen den Kollaborationspartnern zu gewährleisten. Durch den Einsatz bewährter kryptografischer Techniken wird sichergestellt, dass Dateien nicht nur sicher verschlüsselt, sondern auch nachvollziehbar geteilt werden können. Zudem bietet die Unveränderlichkeit des Tangles eine dauerhafte und überprüfbare Dokumentation der wesentlichen Schritte im Austauschprozess.

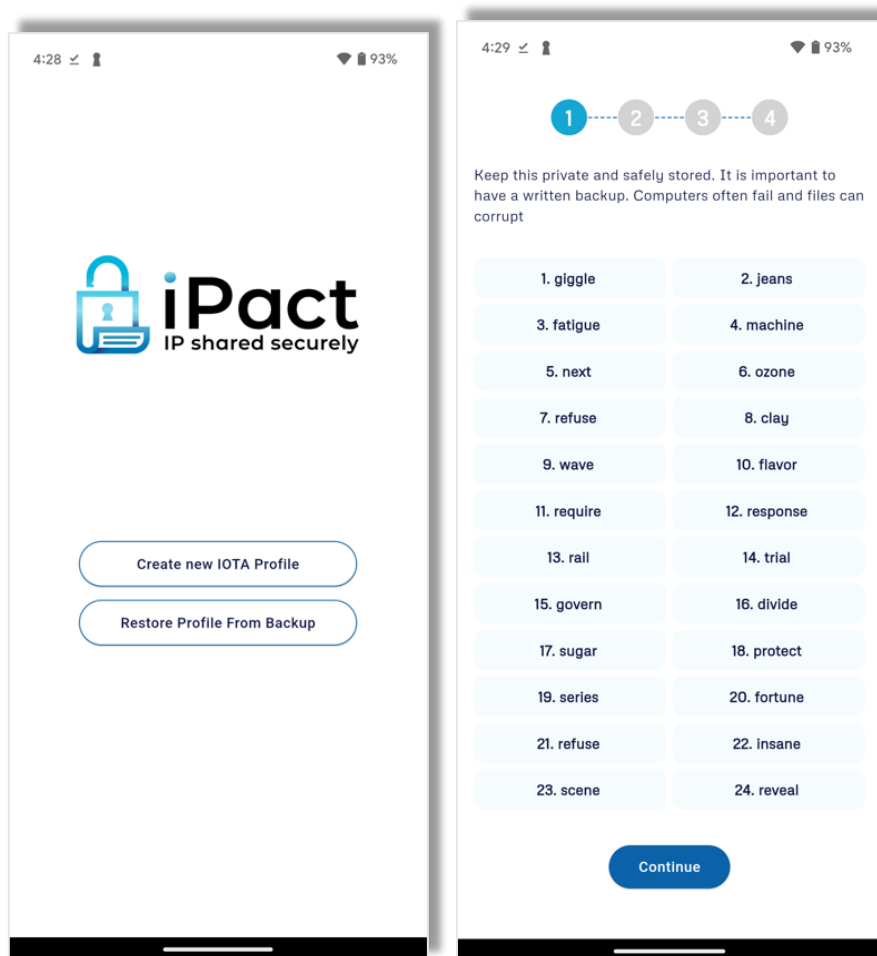
Die Experimente haben nicht nur die technische Machbarkeit des Konzepts belegt, sondern auch eine Grundlage für die Entwicklung einer App geschaffen. Diese Anwendung soll die Funktionalitäten des Frameworks in eine praktikable Lösung für Endnutzer überführt.

5. Praktische Anwendungsergebnisse der iPact-Smartphone-App

Die zentralen Funktionen der Smartphone App iPact umfassen das Herunterladen der App aus dem Google Play Store (eine öffentliche Bereitstellung ist nicht erfolgt, nur für Testnutzer), das Erstellen eines sicheren Profils einschließlich der Generierung einer IOTA-basierten Wallet, die Erstellung von Kollaborationen mittels Einladungen, die sichere Verschlüsselung, Übertragung und Entschlüsselung von Dateien sowie die Überprüfung der Unveränderlichkeit der geteilten Dateien im Zeitverlauf.

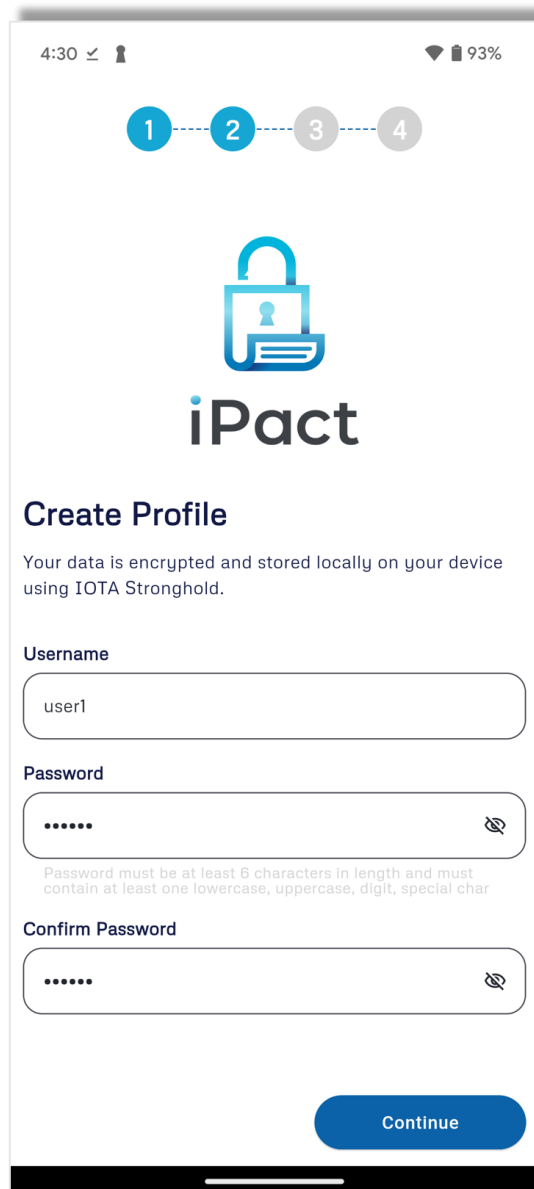
Die Installation und Einrichtung der Anwendung beginnt mit dem Download der App über den Google Play Store. Während der Installation werden Berechtigungen für die Dateispeicherung (zum Hoch- und Herunterladen von Dateien) sowie den Internetzugang angefordert, um die Kommunikation mit dem IOTA-Netzwerk zu ermöglichen. Die Erstellung eines Nutzerprofils kombiniert die Einrichtung eines kontospezifischen Zugangs mit der Erstellung einer sicheren IOTA-Wallet.

Dazu gehört die Generierung eines neuen IOTA-Profiles, das sowohl als Benutzerkonto als auch als Kryptowallet für das IOTA Shimmer Testnet dient. Der folgende links abgebildete Bildschirm ermöglicht Nutzerinnen und Nutzern die Wahl zwischen der Erstellung eines neuen Profils und der Wiederherstellung eines zuvor gespeicherten Profils, z.B. nach einem Gerätewechsel.

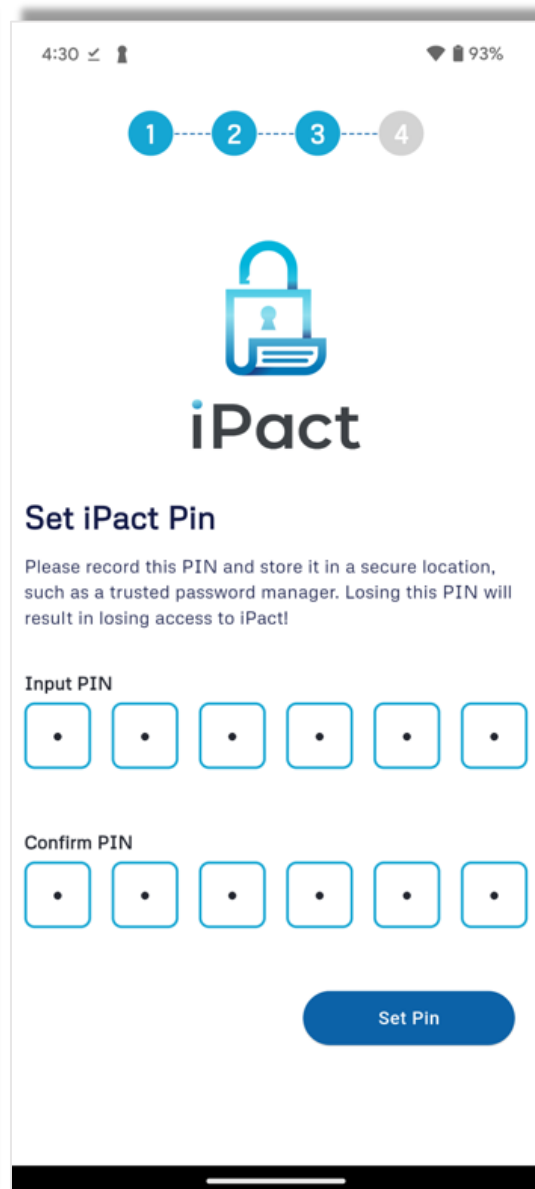


Wahl: bestehendes oder neues Profil 24-Wörter-Mnemonik-Seed Generierung

Der rechte Bildschirm zeigt, wie während der Profilerstellung ein einzigartiger 24-Wörter-Mnemonik-Seed generiert wird. Alle, die schon mal ein Crypto-Wallet verwendet haben, um Kryptowährungen wie Bitcoin, Ethereum o.ä. zu verwahren, kennen diesen Vorgang dem Grunde nach bereits. Der Seed wird sicher innerhalb des Strongholds („lokaler IOTA-Tresor“) der App gespeichert. Die IOTA Stronghold-Technologie schützt diesen Seed vor unbefugtem Zugriff und verschlüsselt ihn mit einem vom Benutzer festgelegten Passwort. Auch eine öffentliche Adresse wird im Rahmen der Profilerstellung generiert, die für Transaktionen genutzt werden kann und über den IOTA Shimmer Testnet Explorer verifizierbar ist.



Profilerstellung/ Stronghold Passwort

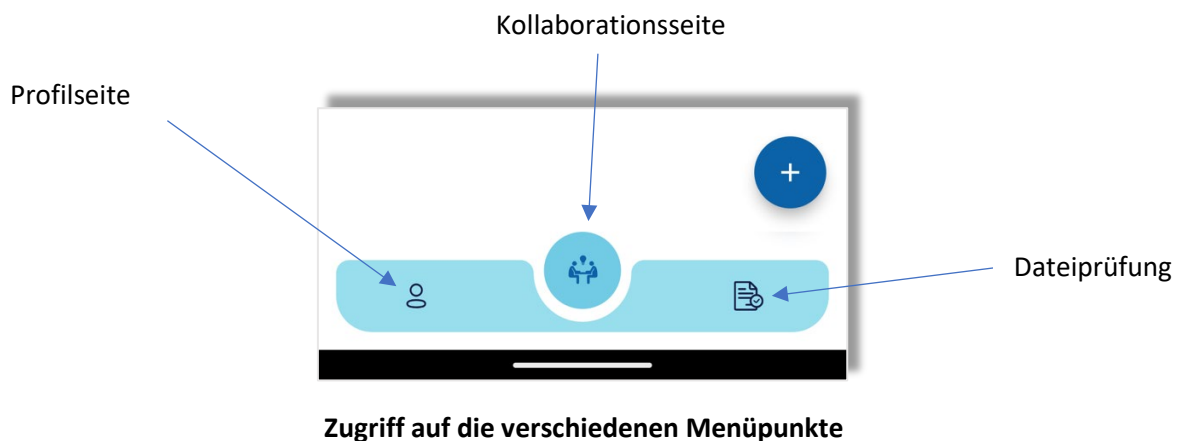


Festlegung des iPact-Pin

Nutzer richten außerdem zwei Zugangsdaten ein: Eine PIN für die Anmeldung in der iPact-Anwendung und ein Stronghold-Passwort, das für Aktionen im Zusammenhang mit dem IOTA-

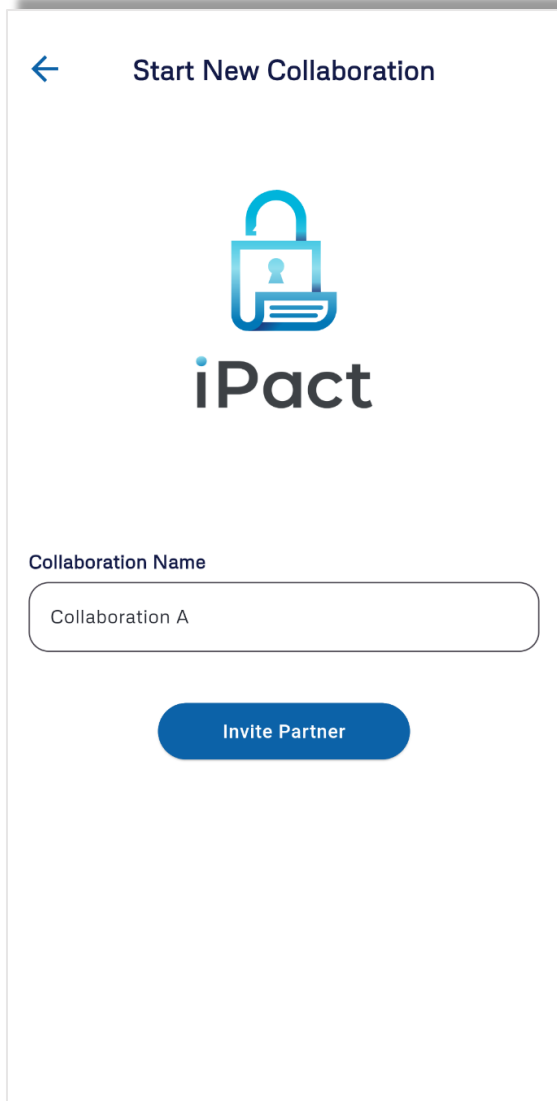
Netzwerk erforderlich ist, wie etwa das Annehmen von Kollaborationen oder das Verschlüsseln und Entschlüsseln von Dateien.

Nach dem Login gelangen die Nutzer zur Hauptseite der iPact-Anwendung, die aus drei zentralen Bereichen besteht: Profilseite, Kollaborationsseite, Seite zur Überprüfung der Authentizität von Dateien. Der folgende Bildschirmausschnitt zeigt diese auf der Unterseite des Screens angebrachten Menüpunkte. Bei diesem Beispiel ist der Reiter aktiviert, der die Kollaborationen zeigt.

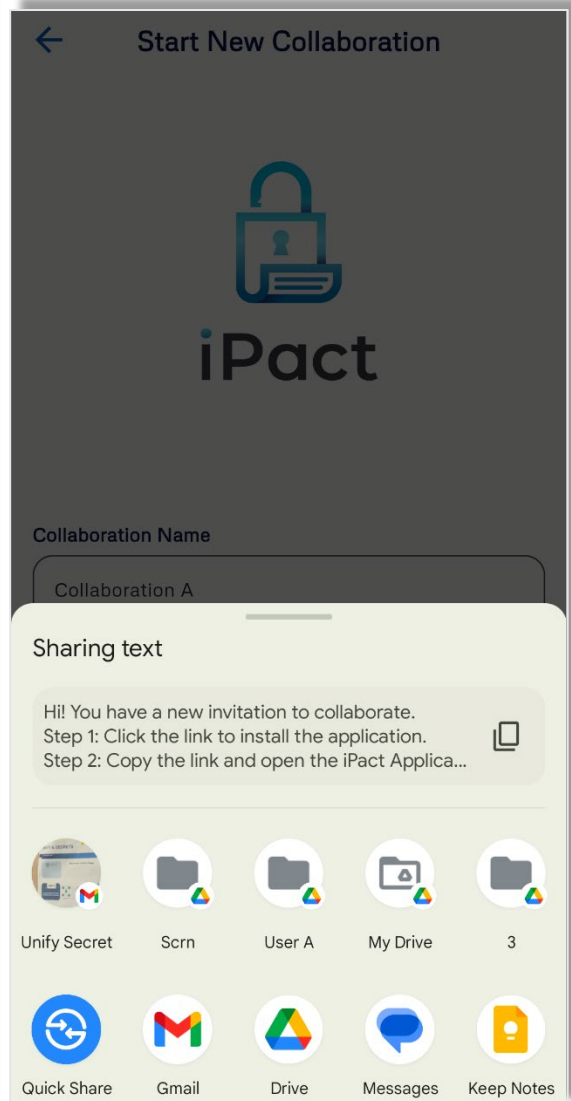


Kollaborationsseite

Das Management von Kollaborationen bildet den zentralen Kern der iPact-Anwendung; hier wird es Nutzern ermöglicht, Dateien sicher zu teilen und zu verschlüsseln.

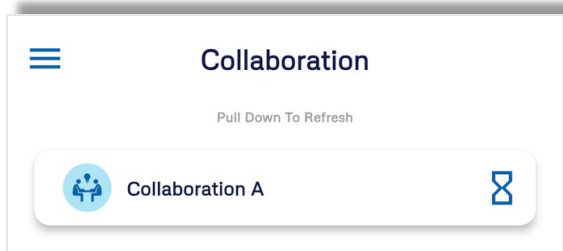


Anlegen einer neuen Kollaboration

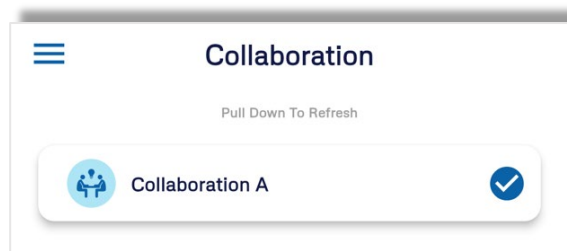


Versenden des Einladungslinks

Bevor Dateien ausgetauscht werden können, muss zunächst eine Kollaboration zwischen den beteiligten Nutzern erstellt werden. Dieser Prozess beginnt damit, dass ein Nutzer eine neue Kollaboration initiiert, ihr einen Namen gibt und einen eindeutigen Einladungslink (einmalig und unverwechselbar) generiert, der dann mit dem gewünschten Partner geteilt werden kann. Der Einladungslink kann komfortabel über verschiedene Kommunikationswege wie E-Mail oder Messaging-Dienste versandt werden. Solange die Einladung nicht akzeptiert wurde, wird der Status der Kollaboration als „Ausstehend“ angezeigt. Sobald der Empfänger die Einladung annimmt, ändert sich der Status auf „Abgeschlossen“, und beide Nutzer können die Kollaboration für den sicheren Austausch von Dateien nutzen.

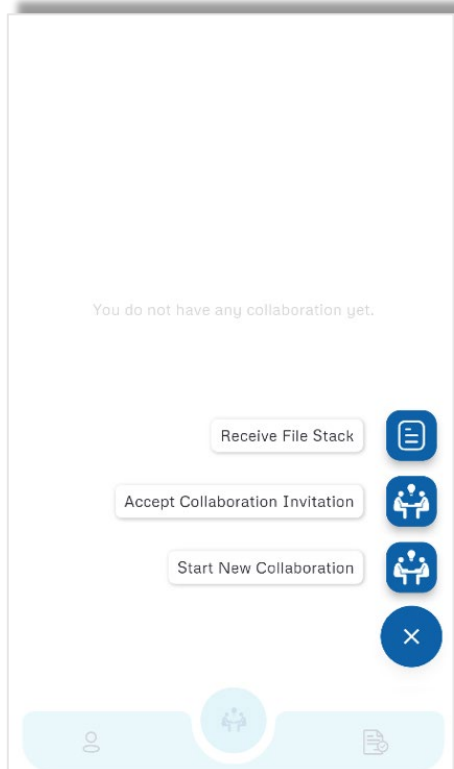


Annahme ist ausstehend

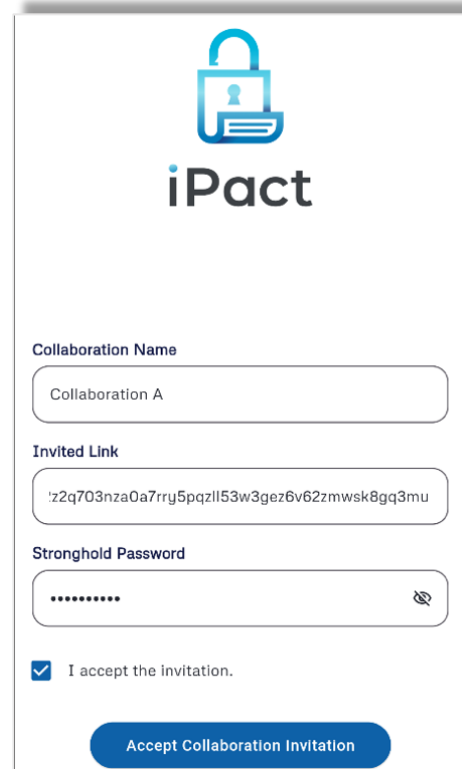


Annahme ist abgeschlossen

Nach Erhalt der Einladung öffnet der eingeladene Nutzer die iPact-Anwendung, akzeptiert die Kollaboration und gibt der Kollaboration einen Namen, der nicht mit der vom Absender gewählten Bezeichnung übereinstimmen muss, hier ist bewusst die Möglichkeit vorhanden, dass jede Partei eine eigene, für sie sinnvolle Bezeichnung wählen kann.



Untermenü „Collaboration“



Benennen und akzeptieren durch Empfänger

Anschließend fügt der eingeladene Nutzer den erhaltenen Einladungslink in das vorgesehene Feld ein und bestätigt die Annahme durch Eingabe seines Stronghold-Passworts. Nach Abschluss dieser Schritte wird die Kollaboration als abgeschlossen markiert, und der Nutzer kann Dateien zur Kollaboration hinzufügen. Parallel dazu aktualisiert der Absender die Kollaborationsübersicht, um sicherzustellen, dass die Kollaboration erfolgreich abgeschlossen wurde.

Collaboration Details

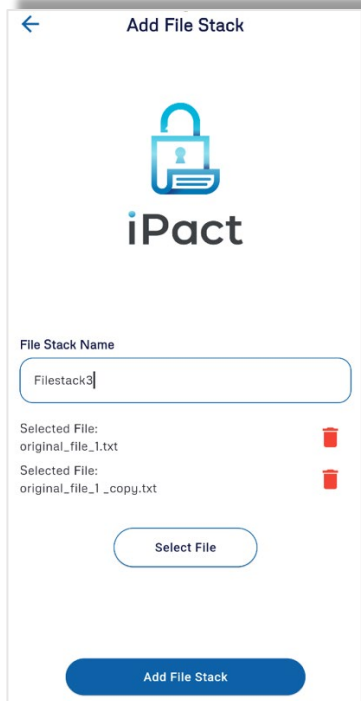
Name: collaboration1
ID: 1726021917617644
Status: accepted
Transaction ID: 0x7809f48c7553ee8901a85d
cf20c55453331fc942a2feaf409248f7ab9529
e8fd
Sender Public Key: MIIBCgKCAQEAE1BSEhXX
qDVnSyxCJLotdEeldjt2vm23u61gbLj74Rhy3s
GdW4Qv3J0ubbj2P/j9qWQEYSXvE1s6TiZBtc
uoKrKRN01aACm6mIOF2tM25ulXT9NmIMm
oHHPmmKkuBpxi9H9bNLEk+VUN+I53abpyz
4Or/UT4qoBU05ed7K+zhCY9VsgYyGsLSoOid
Ukk6bsCwVal5svk9yraU8iqyKrtLMIKjdfB1pq+
D3G3DqnvTKGyW/
0Z5MQRCj6FPJLp+67Jm6XoRW/
nMAScWJXnl4ekUtfZCXu4TeW8rB/02J/5NG
70N2TFgo2OKWKGkSjrfplWZgPsgYHXOD0ut
tnqVYj0pHwIDAQAB
Receiver Public Key: MIIBCgKCAQEAkkh1uu
uEm9iy2CcYuFckVSrHwDK7h8aBiGDzmHZQ
OzzNmOBLjZWw8Z/
1u761V9rhwkZxsWpTdWlZ+XiHw32AdeAM/
pUXrRC0ebVjJfc21glm2DL8i1lec08AZIA39/
Vne9HZQZH3e1VSoxYKmmCvHmai1Y4If6El
/Qv+V9ODDBq1H3ckhoU7C9A46KoKfAByJN
1Y/HUcm5weJlRwpRHYGQh5yJtACHomgs1
Lrrl6vet3klAbXVMq5LmLMFrqoKgshUugheL
++VT7y8jhQeQkXnWSFGp2uiU3Y5VO03Coj
MtSjtU4fh4ejw1pY3nKxOjq3GeF2pz/
ny5WxXBKvQIDAQAB
Sender IOTA Address: rms1qqvee0al7et0p3e
gmjzh8ncflpu3ncdk77nt0yfsr7h9gvavw8kpzv
uepdf
Receiver IOTA Address: rms1qrpjpd7l09a4k
uh0s620fqp9n23rvf44nf8lnpr4zq8wf2aamy5
5hznfw6

[Close](#)

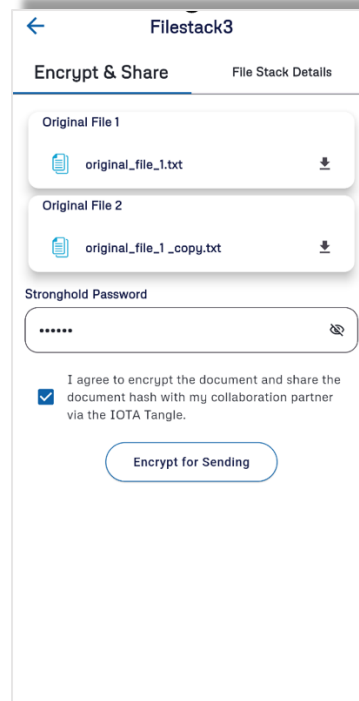
Transparente Beauskunftung der technischen Details möglich

Nutzerinnen und Nutzer können bei Bedarf die technischen Details der Zusammenarbeit einsehen. Mit den IOTA-Adressen ist es möglich, im Tangle Explorer alle Transaktionen zu finden, die mit diesen Adressen verknüpft sind. Diese Adressen dienen als eindeutige Identifikatoren, die Transaktionen im Tangle zugeordnet sind. Über den Explorer können öffentlich gespeicherte Informationen wie Metadaten, Tags oder Hashes eingesehen werden, jedoch keine sensiblen oder verschlüsselten Inhalte, da ohne den privaten Schlüssel keine Kontrolle oder Einsicht in geschützte Daten möglich ist. Dieses transparente Design von IOTA ermöglicht die Nachvollziehbarkeit von Transaktionen, auch in vielen Jahren, selbst dann, wenn iPact als die Applikation, mit der diese Daten auf den Tangle geschrieben wurden, selbst nicht mehr zur Verfügung steht. Auf der Profilseite werden ebenfalls diese wesentlichen Informationen wie die öffentliche IOTA-Adresse und der asymmetrische öffentliche Schlüssel angezeigt.

Sobald die Kollaboration aktiv ist, können Dateien sicher hinzugefügt und ausgetauscht werden. Der Nutzer erstellt hierfür einen sogenannten "File Stack", in den er eine oder mehrere Dateien hochlädt. Jede Datei wird zunächst wie zuvor beschrieben symmetrisch mit AES verschlüsselt. Auf dem IOTA Tangle werden dabei durch iPact zwei Transaktionen vermerkt: eine für den Hash der Originaldatei und eine für den Hash der symmetrisch verschlüsselten Datei.



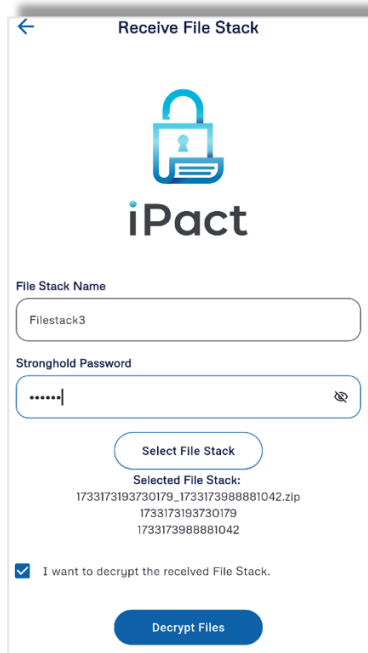
Sender: Erstellen eines File-Stack



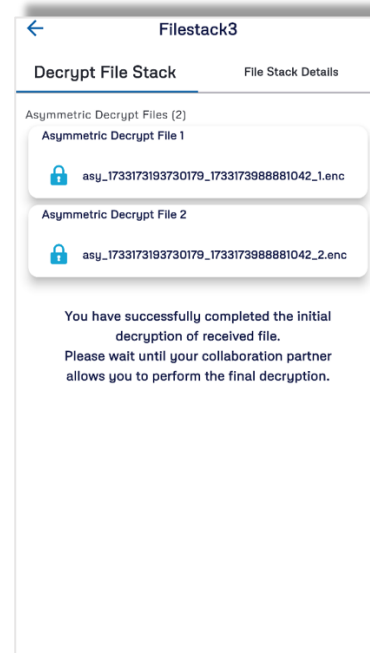
Sender: Verschlüsselung vor Versendung

Nach Abschluss der Verschlüsselung werden die vor Zugriff geschützten Dateien gebündelt und als ZIP-Datei komfortabel, analog zur bereits beschriebenen Versendung des Einladungslinks, über einen herkömmlichen Kommunikationskanal sicher an den Empfänger versendet.

Auf der Empfängerseite beginnt der Entschlüsselungsprozess, sobald die verschlüsselte ZIP-Datei in der iPact-App importiert wurde.



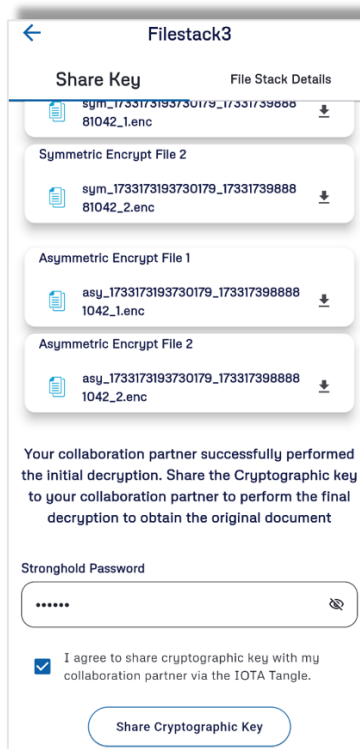
Empfänger: Entschlüsselung 1 Hülle



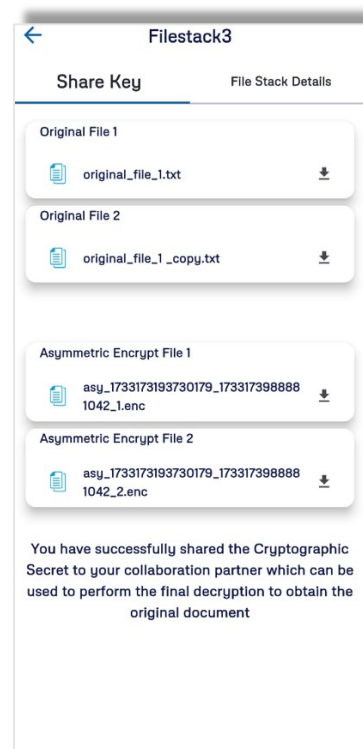
Empfänger: Entschlüsselung 1 Hülle erfolgreich

Der Empfänger entfernt zunächst die äußere Verschlüsselungsschicht mit seinem privaten Schlüssel und speichert die Dateien lokal. Eine Transaktion auf dem IOTA Tangle dokumentiert diesen Schritt

und informiert den Sender über den erfolgreichen ersten Entschlüsselungsschritt, als Voraussetzung für die dann folgende Veröffentlichung des symmetrischen Schlüssels durch den Absender auf dem IOTA Tangle:

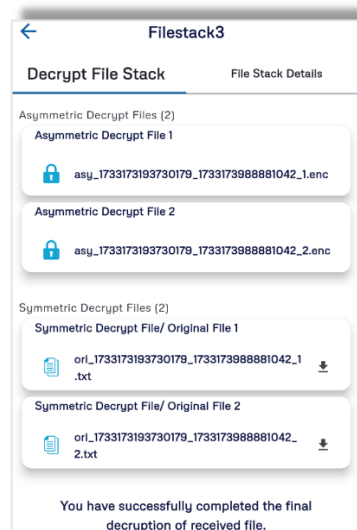


Sender: teilt symmetrischen Schlüssels



Sender: teilen erfolgreich

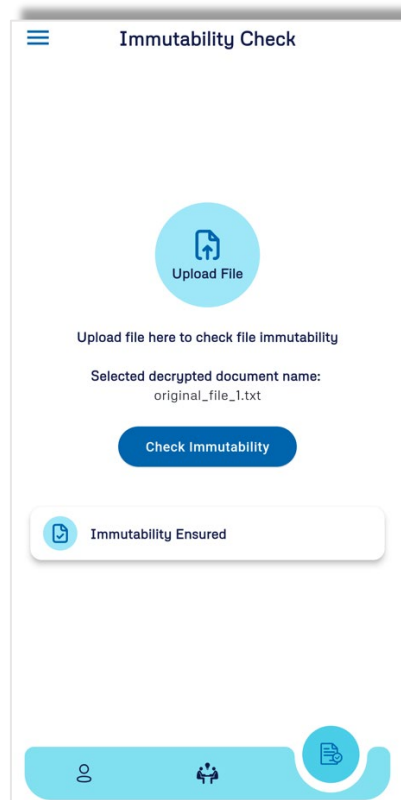
Der Empfänger verwendet diesen Schlüssel, um die verbleibende innere Verschlüsselungsschicht zu entfernen und die Originaldateien vollständig zugänglich zu machen:



Empfänger: finale Entschlüsselung erfolgreich, der Tauschvorgang ist abgeschlossen

Überprüfung der Dateiauthentizität

Der Prozess zur Überprüfung der Unveränderlichkeit dient wie bereits beschrieben dazu, ggf. viele Jahre später, zu prüfen, ob eine geteilte Datei manipuliert wurde oder nicht. Dafür öffnet der Nutzer zunächst die Registerkarte zur Überprüfung der Unveränderlichkeit und lädt die zu überprüfende Datei hoch. Anschließend wird die Funktion „Unveränderlichkeit prüfen“ gestartet, die den Hash der hochgeladenen Datei mit dem auf dem IOTA Tangle gespeicherten Hash vergleicht.



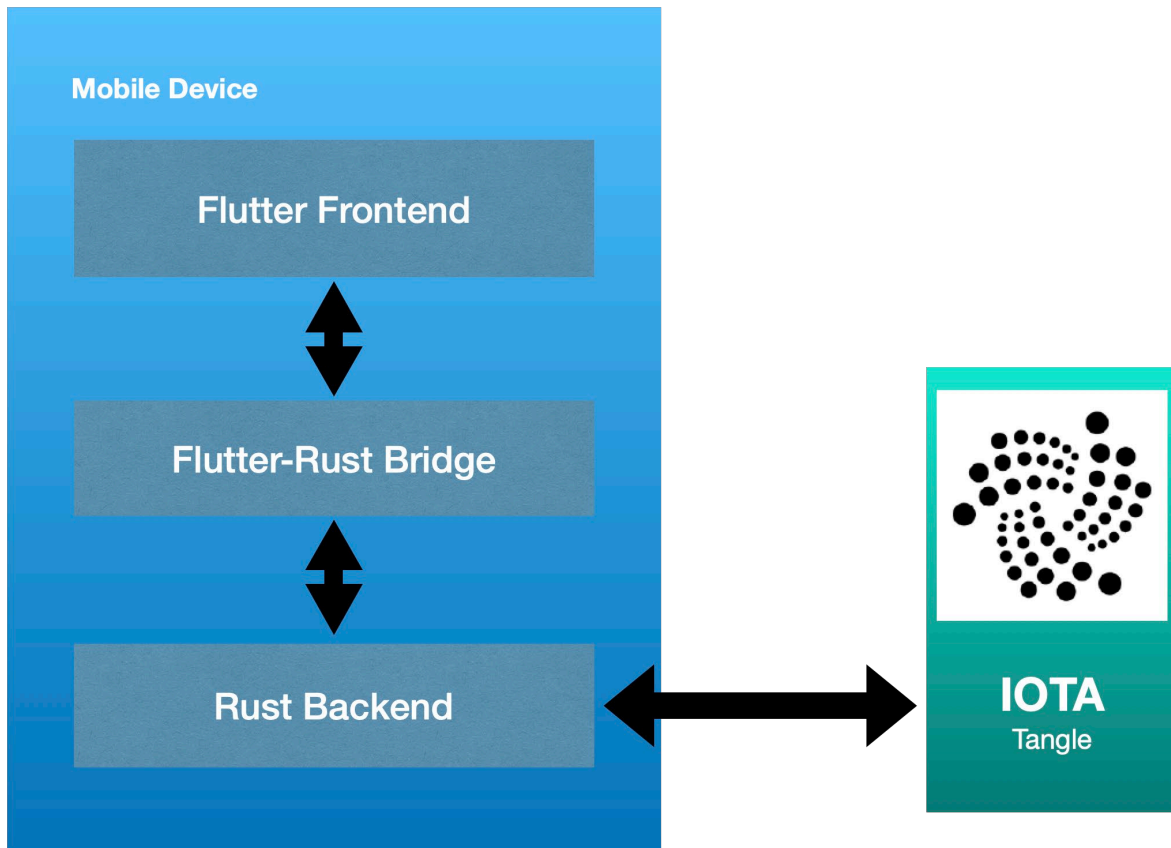
„Immutability Ensured“ – es handelt sich um die Originaldatei

Falls die Datei als unverändert bestätigt wird, zeigt eine entsprechende Nachricht an, dass die Datei unverändert geblieben ist und somit ihre Integrität gewährleistet ist.

Im folgenden Kapitel werden die verwendeten Technologien und Softwarepakete kurz beschrieben, die im Rahmen der Entwicklung verwendet wurden.

6. Überblick verwendete Softwaretechnologie

iPact-Anwendung kombiniert verschiedene Frontend- und Backend-Technologien, dabei spielen neben IOTA die verwendeten Technologien Flutter und Rust eine entscheidende Rolle. Im Folgenden werden die verwendeten Technologien, die Architektur und die kryptografische Funktionalität kurz vorgestellt.



Überblick über die Softwarearchitektur

Flutter, ein von Google entwickeltes Framework, wurde für die Frontend-Entwicklung eingesetzt. Es ermöglicht eine plattformübergreifende App-Entwicklung für Android und iOS mit einer einzigen Codebasis.

Das Backend von iPact wurde in Rust entwickelt, einer Programmiersprache, die ursprünglich von Mozilla initiiert wurde und die besonders für ihre Kombination aus Ressourcensparsamkeit und Sicherheit bekannt ist und entsprechend gut geeignet für sicherheitskritische Anwendungen ist, warum auch die IOTA Stiftung Rust verwendet.

Die Flutter-Rust Bridge verbindet diese beiden Technologien und ermöglicht eine nahtlose Kommunikation zwischen Frontend und Backend. Diese Brücke erlaubt es, komplexe kryptografische Berechnungen sowie Transaktionsoperationen effizient zwischen der Benutzeroberfläche und dem Backend zu übertragen. Der IOTA Tangle fungiert in dieser Architektur als verteiltes Ledger, das Daten wie Transaktionen, Metadaten, Dateihashes und Schlüssel unveränderlich speichert.

Frontend-Implementierung und Kryptografie-Bibliothek

Das Frontend wurde vollständig in Flutter unter Verwendung der Programmiersprache Dart entwickelt. Zu den Hauptfunktionen des Frontends zählen die Registrierung, die Einrichtung sicherer

Profile, der Aufbau von Kooperationen, der verschlüsselte Dateiaustausch sowie die Überprüfung der Unveränderlichkeit von Dateien.

Ein zentrales Element des Frontends ist die eigenentwickelte Kryptografie-Bibliothek `EncryptionUtils`, die Funktionen für symmetrische und asymmetrische Verschlüsselung bereitstellt. Die Bibliothek unterstützt die Generierung von RSA- und AES-Schlüsseln, die Verschlüsselung und Entschlüsselung von Dateien sowie die Integritätsprüfung durch Hashing mit dem Algorithmus SHA-256.

Die Bibliothek bietet umfassende Unterstützung für die Generierung und Verwaltung von Schlüsseln. Dazu gehören Funktionen wie `generateRSAkeyPair`, die RSA-Schlüsselpaar (öffentlich und privat) erzeugt, sowie Funktionen wie `encodeKeyToPem` und `decodePemToKey`, mit denen Schlüssel im PEM-Format sicher gespeichert und wiederhergestellt werden können. Diese PEM-Formate sind weit verbreitet und ermöglichen eine einfache Schlüsselverwaltung über verschiedene Plattformen hinweg. Für die symmetrische Verschlüsselung ermöglicht die Funktion `generateKeyFileForSymmetricCryptography` die Generierung sicherer AES-Schlüssel, die anschließend in einer verschlüsselten Datei abgelegt werden.

Für die Datenverschlüsselung und -entschlüsselung unterstützt die Bibliothek sowohl asymmetrische als auch symmetrische Ansätze. Mit RSA-basierten Funktionen wie `rsaEncrypt` und `rsaDecrypt` können sensible Daten sicher verschlüsselt und nur von autorisierten Empfängern entschlüsselt werden. Die Funktionen `symmetricEncryptFile` und `symmetricDecryptFile` bieten effiziente Methoden für die AES-basierte Verschlüsselung, die besonders für größere Dateien geeignet ist.

Ein weiterer wichtiger Aspekt ist das Dateimanagement und das Hashing. Die Funktion `createHashFromFile` erzeugt SHA-256-Hashes, die als eindeutige digitale Fingerabdrücke von Dateien dienen und die Integrität der Daten sicherstellen. Zusätzlich ermöglichen Funktionen wie `writeEncryptedFile` und `readFileAsBytes` das sichere Speichern und Abrufen von Dateien in verschlüsseltem Format. Diese Funktionen sind speziell darauf ausgelegt, die Verwaltung sensibler Daten sicher und benutzerfreundlich zu gestalten.

Die Kryptografiebibliothek ist vollständig in die Benutzeroberfläche integriert und wird über die Flutter-Rust Bridge mit dem Backend synchronisiert. Dadurch können kryptografische Operationen sowohl auf der Benutzer- als auch auf der Backendentwicklungsebene effizient und sicher ausgeführt werden.

Backend-API-Implementierung in Rust

Das Backend von iPact wurde vollständig in Rust implementiert. Es verwendet das IOTA SDK, um Transaktionen auf dem Tangle zu verwalten, und besteht aus zwei Hauptmodulen: `api.rs` und `wallet_custom.rs`. Das Modul `api.rs` steuert Netzwerkinteraktionen und ermöglicht die Erstellung von erweiterten Transaktionen mit Funktionen wie Zeit-Locks, Metadaten und Speicherdeposits. Das Modul `wallet_custom.rs` bietet erweiterte Wallet-Funktionen, einschließlich der sicheren Verwaltung von Stronghold-Daten und der Erstellung bedingter Transaktionen.

Eine besondere Herausforderung war die Integration neuer IOTA-Bibliotheken wie des IOTA SDK, die frühere Bibliotheken wie IOTA Wallet und IOTA Stronghold ersetzt haben. Diese Umstellung erforderte umfangreiche Tests und Anpassungen, führte jedoch zu einer verbesserten Funktionalität und Sicherheit. Die Flutter-Rust Bridge ermöglichte die effiziente Kommunikation zwischen dem Rust-Backend und dem Flutter-Frontend, sodass komplexe Operationen wie die Erstellung und Verarbeitung von Transaktionen sicher ausgeführt werden konnten.

7. Fazit und Ausblick

Im Zuge der Forschung und Entwicklung von iPact ist im Rahmen des Induko-Projekts eine innovative, dezentrale Applikation entstanden, mit der ein Tausch von Dateien zwischen forschenden Personen und Organisationen manipulationssicher dokumentiert werden kann – ohne eine dritte Instanz, der man sonst vertrauen müsste, wie beispielsweise einem Notar. Durch die Kombination robuster kryptografischer Verfahren mit der unveränderlichen Natur des IOTA Tangle („Blockchain“) bietet dieses Framework eine neuartige, vertrauenswürdige digitale Unterstützung für einen wichtigen Aspekt der kollaborativen Arbeit mit vertraulichen und sensiblen Daten.

Der Einsatz von Dual-Layer-Verschlüsselung, die sowohl symmetrische als auch asymmetrische Methoden nutzt, gewährleistet nicht nur, dass Dateien ausschließlich für autorisierte Benutzer zugänglich sind, sondern ermöglicht durch die innovative Kombination mit den Möglichkeiten von IOTA eine robuste und manipulationssichere Durchführung des sogenannten Handshakes. Die dezentrale Architektur fördert den Peer-to-Peer-Datenaustausch und stellt sicher, dass der Austausch schützenswerter Informationen manipulationssicher dokumentiert wird.

Die Arbeit an iPact war ein Innovationsprozess mit zahlreichen Herausforderungen. Am Ende waren wir sehr froh, nicht nur ein funktionierendes Konzept entwickelt zu haben, sondern auch mit der Umsetzung einer Anwendung als dApp komplett auf dem IOTA Layer 1 zeigen zu können, dass eine solche Lösung sogar benutzerfreundlich möglich ist. Durch die Tatsache, dass der zweite Schlüssel dem Empfänger erst übergeben wird, wenn der Versender zuvor die entsprechende Bestätigung vom Empfänger erhalten hat, entstehen zwar zusätzliche Zwischenschritte, die man beim einfachen Versenden einer verschlüsselten Datei nicht hätte. Dafür erreicht man jedoch die für Beweisbarkeit und Sicherheit entscheidenden Nachweise, ohne eine zentrale Instanz zu benötigen und ohne großen Zusatzaufwand – vor allen Dingen wenn man bedenkt, dass viele Dateien auf einmal ausgetauscht werden können.

Obwohl die gesteckten Ziele grundsätzlich erreicht wurden, gibt es zahlreiche Ansätze für weitere Aktivitäten und Weiterentwicklungen:

- **Umfangreiche Nutzertests und Beseitigung von Schwachstellen**
Ein umfassender Test mit einer größeren Gruppe außerhalb des Kreises der Entwickler war uns innerhalb der Projektlaufzeit nicht möglich.
- **Post-Quantum-Kryptografie**
Um zukünftige Bedrohungen durch Quantencomputer abzuwehren, soll das Framework um quantensichere Algorithmen wie gitter- oder hashbasierte Kryptografie erweitert werden.
- **Erweiterung der unterstützten Datentypen**
Aktuell nur für Textdateien optimiert, sollte die Funktionalität künftig auf Bilder, Videos und komplexe Datensätze ausgedehnt werden. Diese Erweiterung wurde, weil es sich mehr um eine Umsetzungsaufgabe handelt, die nicht zentral für die Kernfunktion ist, zurückgestellt und konnte dann während der Projektlaufzeit nicht mehr umgesetzt werden.
- **Automatisierung mit Flutter Deep Linking**
Manuelle Prozesse wie Einladungen oder Dateitransfers könnten durch Deep Linking automatisiert und benutzerfreundlicher gestaltet werden.
- **Übergang zum IOTA 2.0-Netzwerk**
Die Nutzung des IOTA Shimmer Testnets soll durch das produktionsreife IOTA 2.0-Netzwerk ersetzt werden, das größere Skalierbarkeit, mehr Dezentralität („Coordicide“) und Sicherheit

bieten soll. Nach Projektende wurde im November 2024 jedoch überraschend ein weiterer extrem grundlegender technologischer Wechsel von IOTA angekündigt. Die Plattform soll künftig nicht mehr auf dem Tangle, sondern auf MoveVM und Technologie von SUI basieren. Wie beim IOTA Tangle handelt es sich dabei ebenfalls auch künftig um einen DAG und keine Blockchain; MoveVM wurde von Meta (Facebook) entwickelt und gilt als sehr innovativ und sicher. SUI wurde von ehemaligen Meta-Entwicklern als Weiterentwicklung betrieben, nachdem Meta die Entwicklung einer eigenen Cryptowährung eingestellt hat. Für iPact bedeutet dieser Wechsel, dass die Backend-Entwicklung vollständig auf diese neue Technologie angepasst werden müsste.

- **Multi-User- und plattformübergreifende Unterstützung**

Die Optimierung des Frameworks für mehrere Benutzer (nicht nur 1:1, sondern auch 1:n oder n:n Kollaborationen) und iOS-Geräte.