uID: A Strongly-Secure Usable Identity Ecosystem

1. Executive Summary

This project will theoretically ground, build, and deploy key components of **uID**, a secure, usable, privacy-enabling digital identity ecosystem, able to integrate, and synergize with existing governmental, commercial and open-source identity and authentication solutions.

Designing tomorrow's digital identity solution is faced with unique challenges. Identity mechanisms overwhelmingly refer to and are used by people. They need to be *usable and affordable*, and address individual concerns of *privacy and confidentiality*. At the same time, to ensure trust they need to provide *accountability* and be strongly *secure*. Further, it is important to realize that no one platform can be a sole provider – a viable ecosystem will have standards with well specified APIs and conduits for interoperability that naturally foster a healthy market. Finally, it is essential that these mechanisms interoperate and are efficient so as to not constitute a bottleneck when deployed.

While addressing all of the above challenges, **uID** will focus on two *key* goals: privacy protection and transaction unlinkability. These properties are unfortunately conflicting and require a complex multi-layer research and development approach calling on multi-disciplinary expertise across all the layers of today's digital transactions. Simple "browser plugins" or "email-based" mechanisms alone are bound to fail by not considering the multiple cross-layer security challenges.

The **uID** prototype will be the result of a close collaboration between academic researchers, industry, and digital rights advocacy groups. This guarantees strong trust assurances, marketplace relevance and individuals' privacy protection. Further, **uID** will engage the open-source community early-on as a feedback and development base, to ensure wide community support and acceptance.

uID will demonstrate how to securely integrate existing identity providers, enforce individual privacy, and allow a wide range of users with different technological abilities to access and use their digital identity with ease.

uID will result not only in practical usable software and hardware deliverables, but also in a research knowledgebase composed of reusable protocols, design recommendations, and proposals for standards which will be condensed as entries in a **Trusted Identity Charter**.

The ultimate goal of **uID** is to constitute not only a functional open-source preview of tomorrow's identity ecosystem but also a reference baseline for business and government-driven standards on interoperability, usability, privacy and security in the digital identity space in the years to come.

2. Project Approach

The main overarching goal of **uID** is to ensure accountable privacy and unlinkability. This is why it is important to first understand how identities are used in today's societies and what elements are essential in achieving these assurances.

Consider a typical transaction between an individual (or a software/hardware component with an identity) and a service provider (such as an online website). If the successful completion of the transaction *requires providing intrinsically identifying information* (e.g., IRS tax filing, online purchase of airline tickets etc) neither privacy/anonymity nor unlinkability can be achieved, notwithstanding any assurances of the deployed identity mechanisms. Naturally, the user may choose not to participate in the transaction, but if she does, then the service provider will get access to the identifying information.

Further, if the transaction involves an individual, but no **Trusted Terminal** is available to allow the individual to interact as a client to the service provider, again, privacy and unlinkability cannot be achieved.

Even in the presence of a Trusted Terminal, since underlying network traffic reveals the location and IP address of the service client, **Anonymizers** such as Tor [1] are required to preserve privacy and unlinkability. Yet, current anonymizers do not offer full unlinkability, especially when one considers the entire stack, including application-specific identifying information (e.g., browser or mail client version) that can propagate all the way to the service provider and can identify clients, often with very high accuracy. To mitigate this, **Anonymizers with strong unlinkability** assurances need to be devised, often having to include client-side plugins and logic eliminating or preventing applications' signatures from reaching service providers.

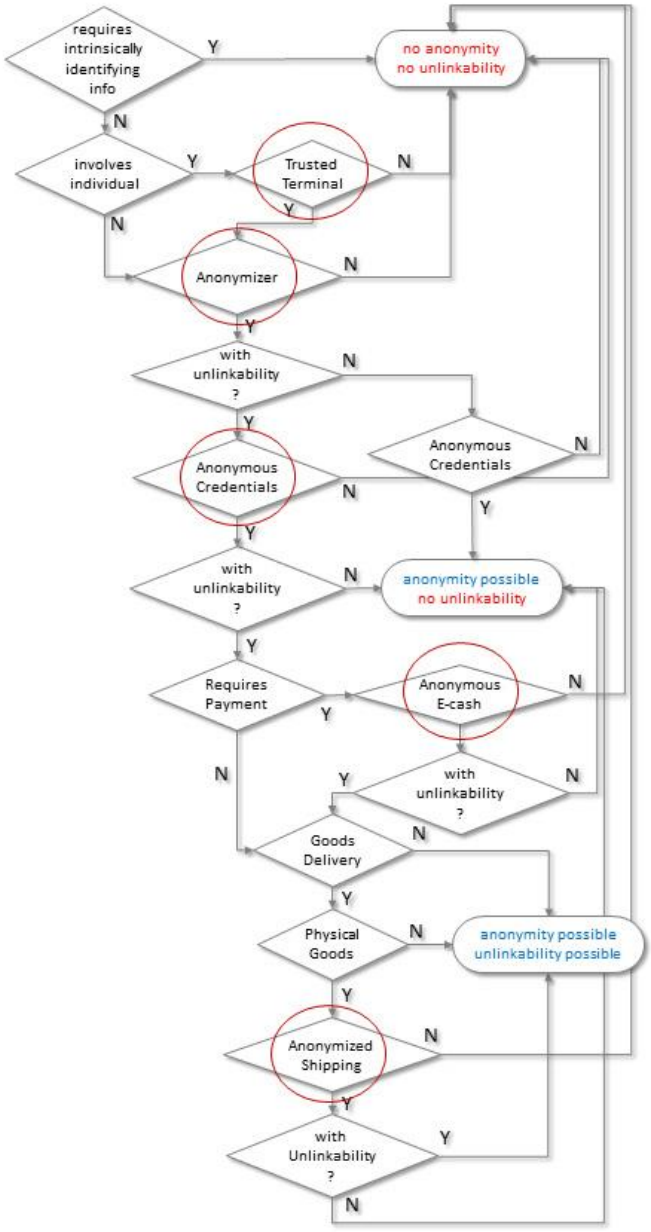


Figure 1. Identity Privacy and Unlinkability can only be achieved through a vertical approach integrating all modern transaction layers. Simple "browser-based" approaches are bound to fail.

Once the terminals and communication conduits are shaped to allow for privacy/unlinkability assurances, **Anonymous Credential** mechanisms are needed to prevent service providers from identifying clients in the authentication and authorization phase, while still allowing them private access to services they are entitled to.

Yet, anonymity is not sufficient to guarantee unlinkability across multiple transactions. This is why it

is important to design and implement **Anonymous Credentials with unlinkability**.

Further, full assurances are not guaranteed, especially if the transaction involves any form of payment, in which case **Anonymous Payment mechanisms with unlinkability** are also required.

Finally, if any goods are to be delivered to a physical address, **Anonymized Shipping** mechanisms will be required – to prevent service providers to directly infer client identities.

2.1. uID Philosophy: A unified cross-layer approach is essential

All of the above components – Trusted Terminal, Anonymizers, Anonymous credentials, Anonymous Payments, Anonymized Shipping – are necessary building blocks in a trusted identity ecosystem with transaction privacy and unlinkability.

This is why overall **identity privacy and unlinkability are not end-to-end solvable**, e.g., by a custom protocol, a "better web browser", a "software plugin" or other user-level tools alone. Instead, they require **a unified vertical approach addressing all the cross-layer privacy and unlinkability aspects**.

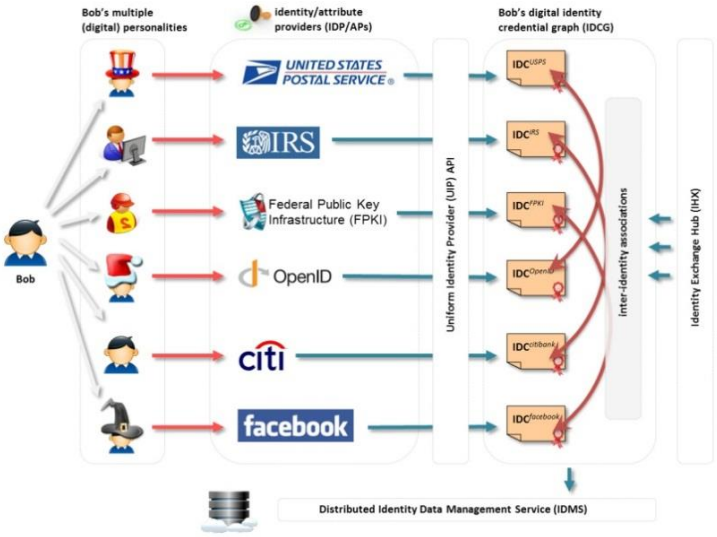


Figure 2. uID seamlessly integrates existing providers. Different identities are aggregated with privacy into an identity credential graph (IDCG) stored in-network by the Distributed Identity Management Service (IDMS).

Unfortunately, a large percentage of today's systems and associated transactions miss these elements, which are essential in achieving a meaningful secure identity ecosystem. **uID** bridges this gap by bringing together experts in privacy, electronic payments, and large-scale systems, in the academic, government, industry and digital rights advocacy communities.

2.2. Guiding Principles

2.2.1. Interoperability

The realization that no single identity solution can satisfy the technological and market needs of a viable identity ecosystem lies at the core of the **uID** vision. **uID** will be designed as an interoperability platform for any existing or future identity solutions. The pilot will demonstrate the integration of Google, OpenID, facebook, and Federal Public Key Infrastructure (FPKI) credentials among others. Further, integration of any new identity technology or platform will be achieved seamlessly by simply providing a short IDC XML meta-description of the platform's identity credentials. Interoperability and extensibility ensure not only wide adoption but also foster innovation and commercial opportunities, while providing structure and synergy in a currently fragmented landscape.

2.2.2. Strong Privacy

The IDC meta-identity will be designed from the ground up to not only integrate arbitrary existing credentials but also provide online pseudonymity as well as full anonymity when desired, on a voluntary, individual-choice basis. Further, the IDC "meta" encapsulator will have the capacity to provide anonymity and need-to-know disclosure (only minimum necessary information shared) *even*

for credentials stemming from legacy providers with no support for privacy! As a result, end-to-end Fair Information Practice Principles (FIPPs) compliance, and strong privacy protection is ensured. The ultimate goal of the privacy controls will be to limit the collection and transmission of information to the minimum necessary to fulfill transactions and their related legal requirements; and to minimize data aggregation and linkages across multiple transactions.

2.2.3. Security and Resilience

uID will integrate with ongoing cyber security advances (several of which are driven by the team's member groups and institutions) and legacy mechanisms. Further, **uID** will also encompass a research and design thrust aimed at eliminating significant vulnerabilities that limit the resilience and security of today's server authentication mechanism. Specifically, in this thrust, novel highly scalable identity credential and authentication paradigms such as the Sovereign Keys concept [41,42] will be taken from the realm of research into a design and implementation phase. This will ensure not only high availability of authentication at scale but also eliminate the numerous vulnerabilities plaguing current PKI-based mechanisms such as SSL/HTTPS. Transactions will be more secure and accountable, identity mechanisms will be more available, and overall trust will increase.

2.2.4. Cost-Effectiveness and Ease of Use

It is virtually impossible to transact in today's highly digital societies without a meaningful minimal set of digital identities. This is why any feasible global-scale digital identity ecosystem will need to provide cheap (basically free in its most basic form), easy to get and easy to use identities. Further, the underlying core identity data management infrastructures are too important to the resilient operation of the identity ecosystem to be left in the care of any single commercial enterprise.

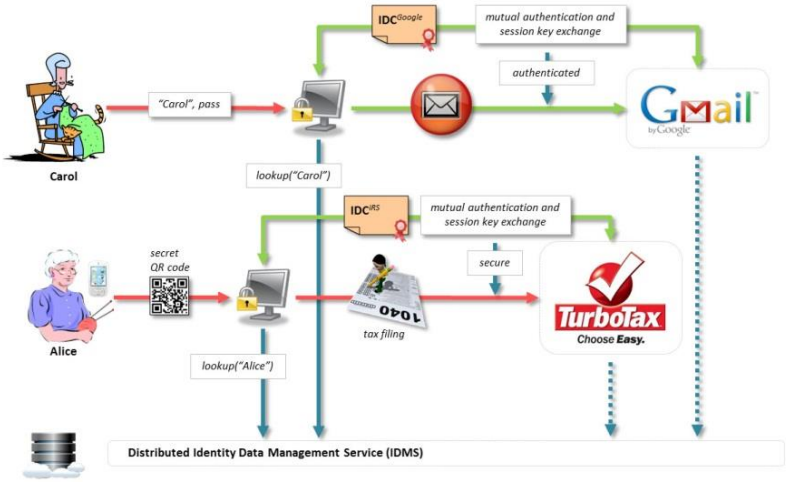


Figure 3. Individuals with different degrees of technological savviness will easily use their digital identity. Arbitrary authentication media and protocols will be supported, including smartphone near field communication, QR-codes, username/password, biometrics and smartcards.

This is why, in **uID**, while identity establishment and authentication will encompass an arbitrary number of providers and commercial/governmental services, the digital identity credential graph (IDCG) records (graph-structured identity and attribute data records of an individuals' multiple identities) will be managed *with privacy assurances* by a distributed identity data management service (IDMS), an in-network, *highly-resilient* service, similar to today's DNS domain name lookup service.

2.2.5. Adoptability

Most importantly, **uID** will **not require service providers to change their current authentication or authorization mechanisms**. We believe this is paramount for wide adoption. The Unified Usable Multi-Identity Manager coupled with the user's IDC encapsulator will deploy new techniques for **collaborative crowd-sourcing** to learn servers' authentication and authorization mechanisms au-

tomatically and allow the user to transact with a single click.

Further, **uID** will **operate across multiple platforms**, including different PC browsers and mobile devices. This is paramount to adoption in a high-tech society in which individuals are mobile, own multiple devices, and need to use different identities in different environments on a daily basis, e.g., at different work places, on the road, and at home.

Finally, **uID** will not lock customers into any single identity provider, but instead act effectively as a cross-provider integration platform.

uID will be not be solely a fundamental research effort but will focus on delivering multiple practical software deliverables that will seamlessly integrate in individuals' lives and run on today's web browsers and smartphones. Further, **uID** will put forward a **Trusted Identity Charter** – a set of concrete actionable proposals for universal identity ecosystem standards. **uID** will be open-source and will engage the community starting with the early design and development stages.

2.3. Fundamental Research Thrusts

In addition to its software deliverables and demonstrations (described later), **uID** will encompass the following fundamental research thrusts:

- A design of strong unlinkability for network anonymizers (Section 2.3.1).
- An Anonymous Identity Credential Encapsulator (IDC) which can integrate arbitrary (including legacy) identity credentials with privacy (and often also unlinkability) guarantees (Section 2.3.2).
- A suite of privacy-preserving, strongly-secure cryptographically-protected payments, identity credential and authentication protocols with anonymity, unlinkability and accountability (Section 2.3.3).
- The proposal of a set of scalable anonymized shipping paradigms (Section 2.3.4).
- The publication of a **Trusted Identity Charter** with concrete proposals for widely applicable core component standards.

2.3.1. Network Anonymizer with Unlinkability

While there is a clear distinction between privacy (protecting personal data) and anonymity (protecting identity), at Internet scale with network-hosted services, behavioral privacy can most likely be achieved only through anonymity, "the state of being not identifiable within a set of subjects." [4]

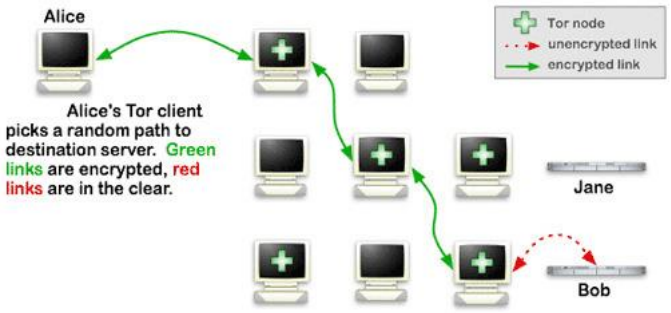
Tor Overview. In the past few decades a flurry of research activity has centered on anonymizing mechanisms in networks, starting with Chaum's Mixnet [5]. Tor is one such anonymizer [1]. Introduced in late 2002, over the past decade Tor has grown to be undoubtedly the most popular low-latency anonymity network for interactive communication.

Tor was originally designed, implemented, and deployed as a project of the U.S. Naval Research Laboratory, for the primary purpose of protecting government communications. Today, it is used every day for a wide variety of purposes by common people, the military, journalists, law enforcement officers, activists, and many others.

Tor is a network of virtual tunnels that allows people and groups to improve their privacy and security on the Internet. It also enables software developers to create new communication tools with built-in privacy features. Tor provides the foundation for a range of applications that allow organizations and individuals to share information over public networks without compromising their privacy.

Using Tor protects against a common form of Internet surveillance known as "traffic analysis." Traffic analysis can be used to infer who is talking to whom over a public network. A basic problem for

the privacy minded is that the recipient of your communications can see that you sent it by looking at headers. So can (un)authorized intermediaries. A very simple form of traffic analysis might involve sitting somewhere between sender and recipient on the network, looking at headers. In more powerful kinds of traffic analysis, attackers spy on multiple parts of the Internet and use sophisticated statistical techniques to track the communications patterns of many different organizations and individuals. Encryption does not help against these attackers, since it only hides the content of Internet traffic, not the headers.



Tor helps to reduce the risks of both simple and sophisticated traffic analysis by distributing your transactions across multiple nodes in the Internet, so no single point can link you to your destination. The idea is similar to using a twisty, hard-to-follow route in order to throw off somebody who is tailing you — and then periodically erasing footprints. Instead of taking a direct route from source to destination, data packets on the Tor network take a random path through several relays that cover tracks so no observer at a single point can tell where the data came from or where it's going.

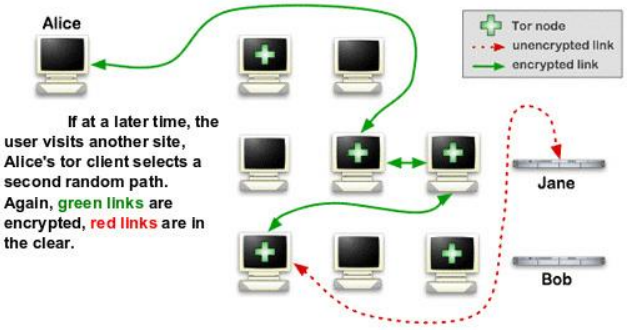
To create a private network pathway, the user's client incrementally builds a circuit of encrypted connections through network relays. The circuit is extended one hop at a time, and each relay along the way knows only it predecessor and successor. No individual relay ever knows the complete path that a packet has taken. The client negotiates a separate set of encryption keys for each hop along the circuit to ensure that each hop can't trace these connections as they pass through.

Finally, like all anonymizing networks that are fast enough for web browsing, Tor does not provide protection against end-to-end timing attacks: if an adversary can observe the traffic at the client computer, and also at the destination, statistical analysis will enable matching with high certainty.

uID and Tor. It is essential to understand that **without in-built network-layer anonymity assurances, no trusted identity mechanism can provide privacy and unlinkability**. This is why scalable low-latency anonymizers such as Tor play an integral in the **uID** vision.

All the core **uID** components will be Tor-enabled by default (users will be able to turn this on/off). Specifically, the Unified Multi-Identity Manager will be designed to transact directly through Tor. uID payments and the IDMS service (see below) will also operate over Tor. Further, the IDMS service access will be fully oblivious and the IDC encapsulators will be designed for full privacy, without containing any client-identifying information.

Tor Unlinkability. For efficiency, Tor uses the same circuit for connections that happen within the same ten minutes or so. Later requests are given a new circuit, to keep adversaries from linking earlier actions to the new ones. This however, still leaves a window of opportunity for adversaries to identify correlations between transactions done by the same client. As part of **uID** we will re-design the Tor random path selection algorithm to achieve strong unlinkability also at transaction-level granularity instead of at fixed temporal granularity. Further, we will pursue a better understanding of how many circuits we should make in theory (and actually make in practice) for



various given client behavior profiles and desired anonymity level. In particular, are there certain client behaviors (like loading particularly ad-filled pages) that would induce an unexpectedly large number of circuits in a short time period? What exactly is the relationship between number of circuits generated and linkability risk? Should we give each isolation class a separate set of entry guard nodes? We should increase the time before we rotate to a new circuit (since ten minutes is quite short if all the streams are known to be from the same session), but what are the tradeoffs between preserving the same website session vs. risks that an attacker traces a long-lived circuit?

A number of related R&D deliverables that we will build are detailed later.

2.3.2. uID Unified Identity Credential Encapsulator (IDC)

uID will act as a cross-provider integration platform, not locking customers into any single identity provider. Central to this capability is the Unified Identity Credential Encapsulator (IDC), a cryptographically-strong mechanism that “wraps” around third party identity credentials.

Sample Healthcare Scenario. To set the stage, consider for example a world where patients’ medical records are finally available digitally. In this case, risks of compromise of patients' privacy increase dramatically. The electronic format makes misuse of many patients' data much easier, so we must be extremely careful with who has access to this data. Consider parties such as insurers and pharmacies that are not actively involved in patient care. Currently, insured patients are required to share the entire record of their medical treatment with their insurer in order to receive benefits, and a pharmacy may store all prescriptions filled for each patient.

However, there is no reason for these parties to see more than an absolutely required amount of enabling information to be able to prevent fraud and verify that the provided treatment is covered under the patient's policy, or that the patient has a valid prescription for the medication being dispensed. We envision a privacy-preserving identity framework to work as follows:

1) Patient sets up an insurance policy with the insurer. The patient will then receive an insured identity credential proving that his treatment should be covered according to the policy.
2) Patient visits doctor/hospital. The patient reveals the relevant part of his policy, and gives the doctor a token for this visit. The doctor/hospital is assumed to be fully trusted by the patient with regard to any record or data generated by that particular visit.
3) Doctor bills insurance company. The doctor generates an anonymous token proving that the insurance claim is valid under the patient's policy and sends it along with a description of the services provided to the insurance company which will check the token and reimburse the claim.
4) Doctor prescribes medications for patient. The doctor uses credentials issued by the state that prove his right to prescribe. The doctor will generate a signed prescription, and an anonymous token showing that the insurance will cover the medication, and transfer both to the pharmacy. He will also generate a token for the patient (potentially as a QR code either printed or uploaded to a smartphone).
5) Patient goes the pharmacy. The pharmacy verifies the tokens it received from the patient and the doctor, then issues the appropriate medications.
6) Pharmacy bills insurance company. The pharmacist combines the token from the doctor and the token from the patient and presents the result to the insurance company as proof of the claim. The insurance company verifies it and reimburses the claim.

Payment for services can thus be achieved without the patient’s identity being revealed to the insurer or pharmacist and without separate visits by the same patient being linkable.

Native uID Anonymous Credentials. Recent developments in cryptography allow this process to happen without revealing any additional information about the patient's record, thus obtaining opti-

mal privacy guarantees. In an anonymous credential system [12,15,17], users can obtain (identity) credentials from an organization, and then when they want to access a resource/service, generate tokens proving that they hold the necessary credentials. Their credential contains a set of attributes, and users are enabled to issue tokens proving that: (a) they have a given attribute, (b) they do not have a given attribute, (c) they have an attribute within a given range, or (d) any combination of such statements. These tokens are anonymous in that they do not reveal any information about the user, they cannot be linked back to the initial issuance, and it is impossible to tell whether two tokens were generated using the same credential.

IDC will feature a "native" anonymous credential built upon our work on the Microsoft Anonymous Health Care System [10] credentials which already feature also the following properties:

- **Delegation.** A user with a credential from an organization can issue a delegated credential to another party. This party will then be able to prove ownership of a credential that was issued by someone with a valid credential from the organization (without revealing information on this intermediary user). The user can also choose which attributes will be included in the delegated credential [11].
- **Single-Use.** In some cases it is important to ensure that no credential is used more than once in the same setting. In this case we require that the user generate a single-use token for each setting - if the user generates 2 tokens for the same setting, it will be easily detected, but as long as each use is in a different setting, there is no way to tell if multiple tokens were generated by the same user [13].
- **Endorsement.** A token can be generated in two parts, such that neither is valid without the other. We call these parts the unendorsed token and the endorsement. The endorsement has the feature that it can be made fairly short, regardless of the length of the statement being proven [16].
- **Revocation of anonymity/Allowing auditing.** In case of audits, the full treatment information and identity for each patient may need to be revealed. To enable this, one option is to have several audit authorities hold shares of a decryption key. When a token is formed to be sent to the insurance company, the doctor can also include the encryption under the corresponding public/encryption key of the full treatment information, as well as her signature on this information. In the case of an insurance audit, the trusted parties can perform the decryption. If fraud is discovered, the doctor can be held responsible. Further, policy revocation can be achieved directly by credential revocation [14].

IDC Wrapper Mode. In addition to native anonymous credentials mechanisms, IDC will also provide a wrapping mode, in which it will naturally integrate any third-party identity credentials. Wrapper mode endows these credentials with confidentiality and stores them as part of the individual's identity graph data in the IDMS service. Further, the **uID** Unified Multi-Identity Manager (discussed later) will access and use these identities. However, note that unlinkability assurances are subject to the properties of the underlying credentials. Some, e.g., using email addresses for authentication (Gmail, Facebook, BrowserID) – cannot offer unlinkability by construction (a costly workaround would be to use separate emails for each transaction). In any case, IDC encapsulators will be designed to inherit unlinkability features (if any) from the underlying encapsulated credentials.

IDCG: Higher Level Data Model Semantics. To structure intra and inter-identity relationships, a scalable data model is required. To this end **uID** will build upon the Higgins project. Higgins is an open source framework designed to integrate identity, profile, and social relationship information across multiple sites, applications, and devices [33]. At IBM Research we have been contributed to the overall design and implementation of the Higgins Framework for years. Specific contributions include the Security Token Service (STS) and associated extensions for SAML and Username Tokens. Additional contributions are planned for Policy Languages, User Interface, and idemix, X.509, and Kerberos Token extensions. And while (as discussed later) Higgins is not privacy-centric, and is particularly unsuited for unlinkability due to centralized trust assumptions, its Perso-

na Data Model is well built to describe and structure identity-centric items. Individual's multiple identities, wallets and relationships will be described using the Higgins Persona Data Model and stored as encrypted IDC graphs (IDCGs) on the oblivious access IDMS service with unlinkability.

Transactions. IDCs will model transactions as challenge-response negotiations. This process will be driven by the Unified Multi-Identity Manager such as to release a controllable amount of information to the transaction peer (e.g., service provider) while keeping the user in control at all times. Naturally, simple sign-on transactions (facebook) will be handled in one round. User control will involve selecting which of its identities – compatible with at least one of the transaction peer's supported authentication mechanisms – will be used.

Finally, the Unified Multi-Identity Manager (discussed later) will deploy collaborative learning to acquire new, previously unknown, challenge-response authentication mechanisms to its capabilities.

2.3.3. uID Payments with Privacy and Unlinkability

Given the importance of online commerce, **any trusted digital identity framework absolutely must provision for secure digital payments that do not compromise the privacy and unlinkability** properties it aims to provide.

2.3.3.1. Anonymous Payments

Unfortunately this is not an easy task to achieve at scale, and is subject to a set of challenges. Since bytes are easily copied, digital cash (electronic payments with anonymity) needs to address the significant issue of accountability (especially in the case of "double spending"). Yet, addressing it via straightforward centralized approaches such as deploying trusted "banks", reduces privacy and unlinkability to direct trust in the banks. Ideally, digital cash should either not allow double spending, or, at the very least incur a penalty for it, e.g., reveal the identity of the double-spender. Also note that digital cash is by definition closely related to digital credentials: it is bound to an individual spender, and it constitutes a proof of qualification (possession of "value").

A large body of research on digital cash has been developed in the past three decades, especially since the introduction of blind signatures by Chaum in 1982 [18]. With very few exceptions, while digital cash has been an interesting research problem, its use has been scarce until recently.

Bitcoin. One recent notable exception is Bitcoin, a fully distributed digital payment paradigm introduced in 2009, which aims to resolve many of the usability and efficiency problems of previous digital cash research efforts, including also mainly the requirement for a centralized bank authority. Although **uID** is agnostic to which anonymous payment system is used, we will consider Bitcoin as a specific approach for our prototype. In Bitcoin, no central authority is required to issue new money and track transactions – now managed collaboratively by the network (of voluntary participants).

Bitcoin deploys public-key cryptography, and its coins contain their owner's public key. When a Bitcoin (BTC) is transferred from Alice to Bob, Alice adds Bob's public key to the coin, and the coin is signed using Alice's private key. Bob now owns the coin and can transfer it further. Alice is prevented from transferring the already spent coin to other users because a public list of all (time-sorted) previous transactions is collectively maintained by the network [19].

The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, thus forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came

from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin at will, accepting the longest proof-of-work chain as proof of what happened while they were offline. [3] Finally, before each transaction, a coin's validity is checked.

Through its inherent elegant simplicity and unassuming security, Bitcoin has arguably spawned a revolution in digital cash. Thousands of real world and online businesses now accept Bitcoins [20], and governments are starting to again consider introducing digital currency – Canada has recently announced its own government-controlled MintChip, which "brings all the benefits of cash into the digital age. Instant, private and secure, MintChip value can be stored and moved quickly and easily over email, software applications, or by physically tapping devices together." [21]

Privacy. Bitcoin was not designed to protect individual privacy. Because transactions are broadcast to the entire network, they are inherently public. Unlike regular banking, which preserves customer privacy by keeping transaction records private, Bitcoin can accomplish a minimal form of privacy, by using different addresses for every wallet (while also having to publish all transactions).

If Alice sends 1.23 BTC to Bob, the network creates a public record that allows anyone to see that 1.23 has been sent from a certain "address" to another. Yet, unless Alice or Bob make their ownership of these addresses publicly known, it is difficult to connect the transaction with them. However, if an adversary links an address to a user at any point they can follow back the transactions as each participant likely knows who paid them and may disclose that information under duress. [22]

Yet, full privacy and unlinkability can be achieved by careful design, incorporating the following:

a) A **Private Bitcoin Meta-Wallet** – an instance of a Bitcoin client integrated within the **uID** client package, used to store individual's Bitcoin ballance. (described in more detail later)
b) Fully anonymized communication conduits using Tor – the Meta-Wallet will automatically deploy Tor underneath to anonymize its network communication.
c) One time use Bitcoin addresses – clients use each address only once
d) The **uID Bitcoin Anonymizer** – **uID** service that relays payments between different addresses and effectively acts as a "laundry", decoupling payers from their payees.

Once relayed through the Bitcoin Anonymizer , a transaction between Alice and Bob is recorded as two transactions between them and the Anonymizer respectively. Naturally, the anonymity pool of this service is formed by all the transactions that it processes at one time, but, given Bitcoin's design and its inherent latencies, timing information can easily be anonymized by including the relayed transactions in different blocks together with other recent transactions (constituting the anonymity pool), at the expense of a slight increase in ultimate transaction confirmation by the Bitcoin network. We believe this is a trade-off between time-to-confirmation and privacy that cannot be refused, especially since immediate transaction confirmation for Bitcoins is most often not essential.

We will also design a version of the Bitcoin Anonymizer which allows audits, e.g., if a critical mass of the Bitcoin network agrees and is willing to participate. This can be achieved, e.g., by having the Anonymizer preserve relayed inter-address links in a format that guarantees a large amount of work required to unlock – e.g. 50% of the Bitcoin's network's capacity. Thus, for serious provable transgressions, enough computation power can be summoned to uncover these links.

Naturally, this raises concerns of powerful parties such as governments' ability to uncover links without asking for network consensus. This is why we will subject audit mechanisms to public scrutiny by the open-source community and establish acceptable anonymity-accountability tradeoffs.

Scalability. Bitcoin currently suffers from a scalability concern in the medium-term. The network is currently rate-limited to 7 transactions per second, mainly to accommodate its flat peer to peer nature. To reach the 5,000 transactions per second peaks that e.g., VISA is currently allowing, Bitcoin will need a few structural changes, mainly centered around distributing the load and the trust hierarchically. Specifically, the intention is to evolve it towards a more typical two-tier structure in which low powered client nodes connect to long-lived, high powered supernodes. As the network scales up, the costs of running a supernode storing full block chain and verifiying every transaction will get progressively higher, but the two tier structure ensures everyone can still get started quickly. [19] We will also briefly investigate several such designs.

2.3.3.2. Alternative Payment Mechanisms

Users may decide to deploy any other type of electronic payments. However, it is important to note that, unless these payments offer privacy and unlinkability, they can and most likely will immediately compromise the assurances provided by **uID**. For example, credit card payments do not provide unlinkability and with very few exceptions, also sacrifice anonymity, since they require identifying information of the card holder. Exceptions are companies offering “anonymous credit card” services, in which a valid (most often virtual) credit card (including number, expiration date and card verification value codes) can be purchased online [23]. Users need to trust the card issuer. Further, prepaid gift/prepaid cards (several of which do not require any identifying information upon activation) can be purchased at convenience stores nation-wide with full anonymity (and sometimes steep “service fees” of up to 16% of their value). Naturally, for unlinkability, different cards would need to be purchased for each individual transaction.

2.3.4. Anonymized Shipping Paradigms

With the increasingly integrated nature of commercial and governmental databases on individuals’ information and behavioral (browsing, buying) patterns, even a single shipment to an identifying physical address can forever compromise the privacy of the associated individual’s identity – by associating the physical address with the individual and the deployed identity — notwithstanding the security of the digital identity that was used in initiating the online transaction. This is why, mechanisms for **Anonymized Shipping with unlinkability** are absolutely essential to the health and sustainability of a trusted identity ecosystem. Unfortunately, they are not trivially achievable.

Anonymized PO Boxes. Consider for example the straightforward idea of trying to protect one’s address by setting up a PO Box with the USPS or a third party, which will potentially also forward the mail to an actual address. This will immediately reveal the link between multiple transactions.

Consider further an apparent fix in which third parties offer “one-time-use” PO Box services. In this case, for efficiency and commercial viability reasons, the actual PO Boxes’ zip code will most likely be relatively close to the actual zip-code of the client, especially for deliveries involving heavier items such as products purchased online. This will immediately reveal correlations and provide increasingly accurate information about the client’s actual identity.

For such PO Box related anonymization techniques to work, the amount of work required to properly anonymize with unlinkability may end up needing to be *linear* in the required level or privacy and unlinkability, an often undesired property that can result in unreasonably high shipping bills. Nevertheless, a number of companies are offering PO Box forward services.

Encrypted Shipping Addresses. Physical PO Box addressing schemes can be augmented with

more sophisticated mechanisms to increase the attained level of privacy. Specifically, we envision mechanisms in which shipping addresses are not released in plaintext during transactions with service providers (websites). Instead, addresses are directly provided in an encrypted format that can be opened only by the shipping courier or a trusted intermediary. Further, the shipping address is decoupled from the actual transaction as much as allowed by the context – i.e., the shipping courier will not know exactly what is being shipped, yet will still eventually know, that the shipment involves, e.g., Amazon.com (and thus infer that the package to be shipped may be a book).

Multiple patents have been filed [24-26] in this arena. Further, at least one existing result [27] based on strong cryptographic constructs (blind, group, and blind group signature schemes) and courier non-collusion assumptions, addresses this problem, and guarantees two essential properties: "the courier company knows at most the merchant or the type of the product shipped, but not the recipient, [and] there is no way for the merchant to recover the address of the intended recipient without collaborating with more than one courier company." We will investigate this and other approaches and create a set of recommendations to be included in the **Trusted Identity Charter**.

2.3.5. Privacy, Commercial Viability and Limits of Unlinkability

Commercial Viability. In any privacy preserving digital identity ecosystem, a subtle trade-off emerges between privacy on the one hand and requirements of accountability and commercial viability on the other. Digital identity privacy assurances need to be carefully designed to consider their cross-market impact and increase their commercial adoption.

Consider the multi-billion dollar online advertisements market, the main driver behind "free" services from thousands of companies such as Google and Facebook. Since the entire market is heavily optimized via and geared for tracking individuals' buying, browsing and general behavioral patterns, introducing unconditional privacy and unlinkability will have a significant impact in the bottom line and operations of this market.

This is why in the medium and long term it is important to provide constructs that also address this aspect, without compromising the core privacy and unlinkability assurances.

Existing frameworks such as Higgins [33] (discussed below) solve this problem by mediating advertisers' requests according to individuals' preferences. Unfortunately, such mediated centralized approaches require trust in the centralized authority and are not designed for privacy or unlinkability outside of this trust assumption.

In a decentralized distributed framework such as uID, instead of today's indiscriminate individuals' tracking mechanisms, we envision targeted, opt-in based mechanisms, in which individuals are anonymously paid (e.g., in Bitcoins) to opt into advertisement networks in which they are exposed to limited, agreed-upon types of ads.

Accountability. Similarly, privacy and unlinkability should not come at the expense of accountability, especially in scenarios involving financial transactions and extreme illicit behavior. This is why in the designed constructs (such as the Bitcoin Anonymizer etc) we will pursue "conditional unlinkability" assurances, in which fraud detection and audits will be possible if, e.g., a critical mass of participants agree that such fraud has been committed.

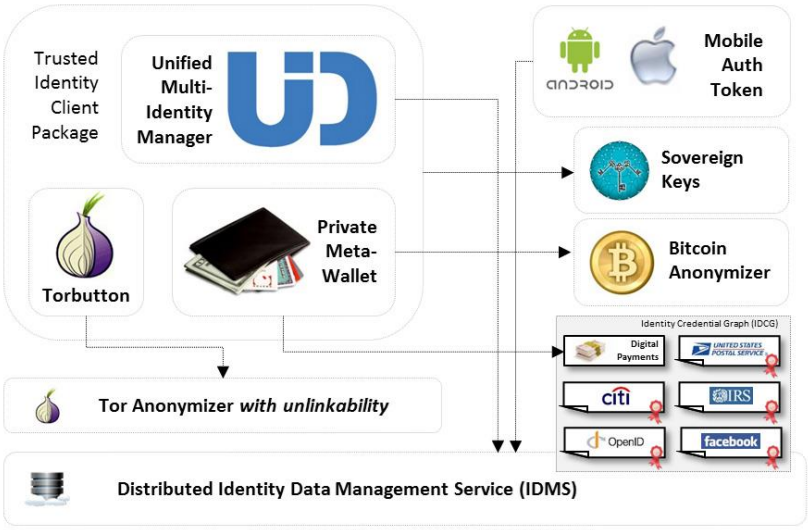
Fundamental Limits of Unlinkability. Note that achieving unlinkability (and anonymity) suffers from fundamental behavior-related limits. Specific, identifiable behavior across multiple transactions will necessarily inter-correlate them despite any security assurances provided by the digital

identity systems deployed. Shopping behavioral studies have shown that more than 50% of all buying is just “simple locating behavior” for familiar brands [28-29]. This naturally yields extremely predictable, highly-correlated shopping carts and thus a certain loss of unlinkability across multiple visits to the same provider. Similar considerations apply to entertainment choices, and entire analytical frameworks have been developed by Amazon or Netflix to bank on this predictability.

2.4. Practical Software Deliverables

uID will result in the following software deliverables detailed further in this section:

- Sovereign Keys: a new strongly-secure, digital identity-specific scalable secure service provider authentication that enables clients to securely assert the identity and trustworthiness of servers.
- An in-network hosted distributed identity data management service (IDMS) for scalable and efficient access to identity records world-wide.
- The uID Bitcoin Anonymizer Service which provides unlinkability for Bitcoin payment chains.



- A Trusted Identity Client Package for Web Browsers and mobile devices, including:
  - The Unified Multi-Identity Manager – an integration platform that will transparently manage a large set of legacy identities and deploy collaborative learning techniques to automatically understand arbitrary new service providers and significantly boost its usability.
  - The Tor Button with Re-enforced Unlinkability – a highly usable “push of a button” version of the popular Tor anonymizer with re-enforced unlinkability guarantees.
  - The uID Secure E-wallet – easy-to-use manager for anonymous unlinkable uID bitcoins.
- Unified Identity Mobile Authentication Token: A smartphone-based identity authentication protocol, as well as a highly *usable* visual (QR-code) and audio based schemes for low-tech individuals.

2.4.1. Sovereign Keys: Scalable Secure Service Provider Authentication

Trusted identity mechanisms can become compromised if clients cannot assert the identity of their transaction peers. Since a vast majority of transactions involve client-server interactions, it is essential to provide secure and scalable server authentication mechanisms.

The mechanism of choice today is SSL, a suite of protocols that provide (mainly public key cryptography based) authentication at the network layer. In web-centric interactions, Unfortunately, SSL suffers from a set of inherent security problems which undermine its usability at large scale. Leaving aside cryptographic protocol vulnerabilities, there are structural ways for its authentication mechanism to be compromised for any domain, through any of the following:

- Break into any Certificate Authority (CA) or compromise the web applications that feed into it. This has been happening with catastrophic results.
- Compromise a router near any Certificate Authority or near a victim’s site, to read the CA's outgoing email or alter incoming DNS packets, breaking domain validation.

- Compromise a recursive DNS server that is used by a Certificate Authority, or forge a DNS entry for a victim domain to defeat domain validation.
- Attack another protocol, such as TCP or BGP, in a way that grants access to victim domain emails.
- A government could order a CA to produce a malicious certificate for any domain. Numerous governments can do this, including some deeply authoritarian ones. Also, governments can easily perform any of the above against other countries' CAs.

In short, as currently implemented, the Web's security protocols may be good enough against attackers with limited time and motivation, but cannot defend from seriously incentivized parties.

At USENIX Security 2011 we have reported a number of findings that came from the SSL Observatory [41]. As of October 2011, 248 certificate compromises had been issued by 14 CA organizations. **Each of these incidents could have broken the security of any and all HTTPS websites**.

The problems of TLS authentication are urgent and structural, but they can be fixed. To this end we have initiated the Sovereign Keys project, which aims to make authentication at Internet scale more reliable and secure. It will fix structural insecurities in the way that the Web, Email and other Internet protocols currently establish encrypted connections, and will protect HTTPS and other uses of TLS/SSL against a wide variety of attacks, including attacks involving Certificate Authorities and domain validation, and attacks that involve downgrading or blocking encrypted connections. In **uID**, Sovereign Keys will provide strong authentication of server identities, ensuring that there are no unavoidable third-party points of attack.

Sovereign Keys operates by providing a highly secure way of associating domain names with public keys, augmenting other methods of publishing TLS/SSL keys, such as the existing system of CAs. The design allows clients and servers to use cryptographic protocols without having to depend on any third parties after the moment the server creates a Sovereign Key.

The design also aims to remove client-side in-browser certificate warnings altogether, and to replace them with automatic circumvention of attacks. This is important because research has shown that human beings don't understand certificate warnings and very often click through them. An example of this problem is the man-in-the-middle attack that was observed by Syrian Facebook users in May of 2011. That attack – conducted with an arbitrary certificate, not signed by a trusted CA – so every target would have seen a warning message. However, research [6] indicates that a large proportion of those targets clicked through and logged into their Facebook accounts anyway.

Sovereign Keys is based on a semi-centralized, verifiably append-only data structure. This is in some respects similar to the Bitcoin protocol, although it relies on CA-signed certificates, rather than proofs of cryptographic work or "mining", as a rationing mechanism. Master copies of the append-only data structure are kept on a small number of 10-20 "timeline servers". The level of trust that must be placed in them is very low, because the Sovereign Key protocol is able to cryptographically verify their functions. Sovereign Keys are preserved so long as at least one server has remained good. For scalability, verification, and privacy purposes, copies of the entire append-only timeline structure are stored on a set of "mirrors".

Clients learn about Sovereign Keys by sending (encrypted) queries to mirrors. Once a client knows a Sovereign Key for a domain, that fact can be cached for a very long time, with only occasional queries to check for revocations. This can make the protocol quite robust even if mirrors are malicious, blocked, or just unreliable. Clients can keep using the protocol for long periods under very hostile network conditions (like those you might find in Syria, Iran or Burma).

In **uID** Sovereign Keys integrate with the Trusted Identity Client for strong server authentication.

2.4.2. Distributed Identity Data Management Service (IDMS)

The usability of any identity ecosystem is directly related to the amount of work required to access and utilize one's identity. It is thus important not to burden users with unnecessary direct management and storage of identity credentials, but rather minimize the number of tokens they must possess. Ideally, users should be empowered to walk up to any terminal, present an easy-to-carry credential (e.g., a username/password pair, a printed QR code, or a mobile device) and gain access to any of their (multiple) digital identities.

To enable this, **uID** maintains identity data in the Distributed Identity Data Management Service (IDMS), a separate, high-availability, distributed data management layer run by a large, loosely connected set of nodes, a subset of the Tor anonymizer network. Not unlike DNS – which maps host names to IP addresses – the IDMS allows terminals under the control of users to retrieve records associated with their identities, including their IDC graphs, Bitcoin wallets etc.

uID and Higgins. IDMS bears certain similarities to the "Personal Data Service" (PDS) found in the Higgins project [33] discussed above. The Higgins architecture however is not designed for privacy and is particularly unsuited for unlinkability assurances; it is based on a centralized model relying on a complex, trusted support infrastructure. Clients and individuals are forced to entrust their privacy to this infrastructure and different accesses for a given identity's records are directly linkable at all layers, including the network, the Higgins core, and applications.

The **uID** IDMS service will be built from the ground up with privacy and unlinkability assurances without requirements of centralized trust. IDMS will feature **full access privacy**. Clients will be able to **obliviously** access their identity data – without revealing inter-access correlation, not even to the IDMS layer itself. This guarantees strong transaction unlinkability. We will build IDMS on the open-source Apache Cassandra distributed database management system [40]. Cassandra is an excellent fit for our requirements because it is designed to handle very large amounts of data, spread across many servers, while tolerating multiple servers' failures. Cassandra is widely used by companies including Netflix and Rackspace to serve hundreds of millions of users. To achieve access privacy we will redesign the Cassandra codebase to implement our latest, state-of-the-art oblivious data access protocols, including the fastest-to-date ORAM mechanisms [34-39].

2.4.3. Bitcoin Anonymizer Service

The Bitcoin Anonymizer is a service that relays Bitcoin transactions between parties. In doing so, it eliminates direct links between payment endpoints and increases a monitoring adversary's uncertainty with respect to the set of parties transacting within close time proximity. Since immediate transaction confirmation is not provided nor essentially required by Bitcoin's eventual consistency model, clients can specify longer anonymization delays or desired anonymity set sizes. Upon receiving an inbound payment from Alice, the Anonymizer will wait for either the specified time interval to pass, or for the required number of additional transactions before relaying the received payment to its recipient, Bob.

Naturally, the anonymity pool of this service is formed by all the transactions that it processes at one time, but, given Bitcoin's design and its inherent latencies, timing information can easily be anonymized by including the relayed transactions in different blocks together with other recent transactions (constituting the anonymity pool), at the expense of a slight increase in ultimate transaction confirmation by the Bitcoin network.

For increased security, the Bitcoin Anonymizer can be distributed across multiple dedicated nodes, or by organizing all Bitcoin clients in a peer-to-peer anonymization infrastructure. We will also design a version of the anonymizer which allows audits, e.g., if a critical mass of the Bitcoin network agrees and is willing to participate. We will implement this by having the Anonymizer preserve relayed payment information in a cryptographic format that guarantees a large predictable amount of work required to unlock – e.g., larger than 50% of the current Bitcoin's network's capacity. Thus, for serious provable transgressions, enough computation power can be summoned to uncover fraud.

2.4.4. Trusted Identity Client Package for Mozilla Firefox and Android

The central, client-facing **uID** interface is the Trusted Identity Client Package, an open-source software package for both web browsers and Android which will interface with the **uID** core services to allow individuals to manage all their different digital identities, and payment mechanisms with full privacy and unlinkability controls. The client package contains a number of specific tools, including the Unified Multi-Identity Manager, the Bitcoin Meta-Wallet, and the anonymizer controls exposed through the Tor Button with re-enforced unlinkability.

2.4.4.1. Unified Multi-Identity Manager *with crowd-sourced learning capability*

The Unified Multi-Identity Manager is a client-side software component that empowers users to integrate and use multiple identities and wallets seamlessly, while preserving their privacy and unlinkability properties. The Manager will be based on the open-source Higgins Active Client, which will be completely re-written for privacy and unlinkability. It will integrate natively with the Tor anonymizer, and connect to the Sovereign Keys infrastructure, the uID Bitcoin Anonymizer, and IDMS service with full access privacy. Further, we will endow the Manager with collaborative learning abilities to acquire new, previously unknown, challenge-response authentication mechanisms to its capabilities. To this end, across a number of initial instances, the Manager will observe users authenticate manually (e.g., for unknown websites). In the process it will learn with increasing confidence that, e.g., the typed-in string corresponds to the username website text-field, or to the OpenID URI, both of which can be found in the individual's IDCG. After a number of manual authentication processes have been observed by different Manager instances running globally, they will exchange and agree upon information representing the authentication workflow and its relationship to the individuals' data models. In any future interactions, individuals will then be offered the choice of automatic login. The Manager will interrupt users only to expose necessarily interactive authentication steps such as in the case of transactions requiring proof of being human (such as visual CAPTCHAs or puzzles). In effect, users will train the system via crowdsourcing.

2.4.4.2. Tor clients with Re-enforced Unlinkability

Tor: improving isolation in the browser. As discussed above, Tor can't solve all anonymity problems. It focuses only on protecting the transport of data. Software-specific data such as browser type and configuration can provide just enough identifiable information to compromise privacy.

While the original Tor design was completely application neutral (meaning it treated TCP streams as opaque and focused entirely on anonymizing the source and destination IP addresses), it quickly became apparent that normal users need assistance with application-level privacy issues too. We developed the Torbutton Firefox extension [30], and later moved to Tor Browser, a fully self-contained software bundle including a forked version of Firefox [31].

The Tor Browser design aims to provide seven security and privacy properties: proxy obedience, state separation, disk avoidance, application data isolation, cross-origin identifier unlinkability, cross-origin fingerprinting unlinkability, and long-term unlinkability. We will continue to work with the Mozilla security team to identify and help resolve bugs and design flaws in Firefox that prevent us from keeping our users safe. Lastly, we're working with Mozilla and other browser vendors to help them design an effective "anonymous browsing mode" – the current private browsing mode does not include a network adversary or websites (including ad networks) in its threat model [32].

Tor: separate streams by tab. Tor faces a trade-off between privacy and scalability: on the one hand, we can maximize isolation by giving every new application request its own Tor circuit. On the other hand, because the process of building circuits requires Tor relays to perform public key operations, the current relay pool likely can't handle the load of all users making far more circuits than they do now. Further, splitting application requests too much threatens both anonymity (the more circuits the user makes, the greater the risk that one of them is observable by an adversary) and usability (many websites can't handle user sessions with fetches coming from different IPs). One balancing point is to group all application requests from a given browser tab onto the same circuit. The session behavior is naturally preserved, and different sessions are naturally isolated. Since, unfortunately, Firefox still makes it very difficult to learn which requests come from which tabs, we will approximate this by isolating requests based on the domain in their Referer header. Tor has now experimental support for applications to specify isolation classes for new streams.

Torbutton for Android. Given the pervasiveness of smartphones in today's societies, it is paramount to enable privacy and unlinkability properties, and thus allow cross-platform secure use of digital identities. We will start by tackling the main issue of network and client anonymization via Tor. Currently the Tor port to Android uses a browser called Orweb, with a simple proxy-setting add-on called Proxy Mobile. It includes no application-level privacy, nor any isolation features.

The non-mobile Torbutton plants a wide variety of request observers deep inside Firefox, to intercept potentially dangerous behavior and neutralize it as needed; these hooks are not so straightforward in Firefox mobile's multi-process model. We will investigate two alternative approaches going forward: (i) adding Torbutton-like features into Proxy Mobile, with the eventual goal of having Proxy Mobile replace Torbutton on all platforms, and (ii) re-compile Firefox mobile's source with all the proper defaults needed as the Android Tor Browser.

2.4.4.3. Private Meta-Wallet

The Private Meta-Wallet is a client software that manages clients' multiple payment mechanisms. In its first release the Meta-Wallet will be effectively a Bitcoin client redesigned to provide anonymous payment guarantees, generate new payment identities on-demand, and integrate with Tor and the Bitcoin Anonymizer. The wallet data model is stored encrypted as part of a client's identity records in the IDMS which the Meta-Wallet can access obliviously.

2.4.5. Unified Identity Mobile Authentication Token App

The uID Unified Identity Mobile Authentication Token will be a smartphone application that allows individuals to easily authenticate at the touch of a finger, to each other and to other networked terminals using any of their multiple identities.

A core challenge that we will address will be the seamless setup of secure channels with other individuals and their mobile tokens as well as with merchants' terminals and banks. These channels

will be set up efficiently while preserving user verifiability, privacy, and existing device compatibility.

The Authentication Token will extend the codebase of two systems we have been working on since 2004, namely SiB [43] and SafeSlinger [2]. Both systems rely on a novel approach converting physical trust into digital trust. Human users can leverage human-verifiable physical aspects to establish trust in a digital process, such as an online service or a digital communication link.

SiB makes use of camera-endowed smartphones to visually acquire public keys of other entities. Since users can visually see which entity they establish a key with, human verifiability is enabled. SafeSlinger (a fully functional currently downloadable iPhone/Android application) utilizes state-of-the-art cryptographic constructions to exchange credentials among multiple mobile devices, without requiring physical co-location. SafeSlinger establishes a secure channel offering secrecy and authenticity, by essentially, safely "slinging" information from one device to another. SafeSlinger also provides an API for importing applications' credentials into a user's world model.

The **uID** Authentication Token app will allow users to securely store their IDCGs or a selected subgraph thereof and authenticate to peers (other tokens, trusted terminals, or merchants' counters) through both near field communication (Bluetooth, WIFI) or visual channels (QR and bar code). The app will also be able to obliviously connect to the IDMS service through Tor and access identities' data and payment credentials with the scan of a fingerprint.

Adversaries that gain illicit access to the Mobile Token may attempt physical attacks to retrieve stored credentials. We will investigate how to deploy the new IBM SecureBlue++ project's trusted hardware deliverables to significantly increase defenses against such physical attacks.

References

[1] The Tor Project, online at https://www.torproject.org/

[2] Michael Farb, Manish Burman, Gurtej Singh Chandok, Jon McCune, Adrian Perrig, "SafeSlinger: An Easy-to-use and Secure Approach for Human Trust Establishment" and "SafeSlinger App for mobile devices", online at http://www.cylab.cmu.edu/safeslinger/

[3] Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System", January 2009

[4] PFITZMANN, A., DRESDEN, T., AND HANSEN, M. 2005. Anonymity, unlinkability, unobservability, pseudonymity, and identity management a consolidated proposal for terminology.

[5] CHAUM, D. 1981. Untraceable electronic mail, return addresses, and digital pseudonyms. Communications of the ACM 24, 2.

[6] Joshua Sunshine, Serge Egelman, Hazim Almuhimedi, Neha Atri, and Lorrie Faith Cranor, Carnegie Mellon University, Crying Wolf: An Empirical Study of SSL Warning Effectiveness, USENIX Security 2009

[7] BAUER, K., MCCOY, D., GRUNWALD, D., KOHNO, T., AND SICKER, D. 2007. Low-resource routing attacks against tor. In Proceedings of the 2007 ACM workshop on Privacy in electronic society. WPES '07. ACM,New York, NY, USA, 11–20.

[8] DINGLEDINE, R. AND MATHEWSON, N. 2006a. Anonymity loves company: Usability and the network effect. In Proceedings of the Fifth Workshop on the Economics of Information Security WEIS 2006.

[9] UNITED NATIONS. The Universal Declaration of Human Rights. Online at http://www.un.org/en/documents/udhr/.

[10] Melissa Chase, Kristin Lauter: An Anonymous Health Care System. IACR Cryptology ePrint Archive 2011: 16 (2011)

[11] M. Belenkiy, J. Camenisch, M. Chase, M. Kohlweiss, A. Lysyanskaya, and H. Shacham. Randomizable proofs and delegatable anonymous credentials. Crypto 2009.

[12] S. Brands, "Rethinking Public Key Infrastructure and Digital Certificates:Building in Privacy", PhD thesis, 1999.

[13] J. Camenisch, S. Hohenberger, M. Kohlweiss, A. Lysyanskaya, and M. Meyerovich. How to win the clonewars: efficient periodic n-times anonymous authentication. CCS '06.

[14] J. Camenisch and A. Lysyanskaya. Dynamic accumulators and application to efficient revocation of anonymous credentials. Crypto '02.

[15] J. Camenisch and A. Lysyanskaya. A signature scheme with efficient protocols. SCN '02.

[16] J. Camenisch, A. Lysyanskaya, and M. Meyerovich. Endorsed e-cash. IEEE Security and Privacy '07.

[17] D. Chaum. Security without identification: Transaction systems to make big brother obsolete. Communications of the ACM, 28(10):1030{1044, Oct. 1985.

[18] David Chaum, Blind signatures for untraceable payments, Advances in Cryptology - Crypto '82, Springer-Verlag (1983), 199-203.

[19] Bitcoin Wiki, online at https://bitcoin.it/

[20] "Businesses that accept Bitcoin", online at https://en.bitcoin.it/wiki/Trade

[21] "The MintChip Challenge", online at http://mintchipchallenge.com

[22] "Bitcoin Wikipedia Page", online at http://en.wikipedia.org/wiki/Bitcoin

[23] Ryan Barrett, "Privacy through prepaid credit cards", online at http://snarfed.org/privacy_through_prepaid_credit_cards

[24] Stephen Christopher O'Donnell et al, US Patent 2002/0013739 A1, "Apparatus and method for providing anonymous shipping services"

[25] First Data Corporation, "Anonymous mailing and shipping transactions", US Patent number: 7213748

[26] Hall Aluminum LLC, "Method and apparatus for masking private mailing address information by manipulating delivery transactions", US Patent number: 7240035

[27] Elli Androulaki, Steve Bellovin, "APOD: Anonymous Physical Object Delivery", Privacy enhancing technologies: 9th international symposium, PETS 2009.

[28] W. Wells, L. Losciuto, “Direct observation of purchasing behavior”, Journal of Marketing Research 1966

[29] M. Sutherland & T. Davies, 'Supermarket shopping behavior: An observational study', Caulfield Institute of Technology Psychology and Marketing Series, 1978

[30] Torbutton, online at https://www.torproject.org/torbutton/en/design

[31] TorBrowser, online at https://www.torproject.org/projects/torbrowser/design

[32] Mozilla Anonymous Browsing, online at https://wiki.mozilla.org/Security/Anonymous_Browsing

[33] The Higgins Project, online at http://www.eclipse.org/higgins/

[34] Martin Franz, Peter Williams, Bogdan Carbunar, Stefan Katzenbeisser, Radu Sion, "Oblivious Outsourced Storage with Delegation", Financial Cryptography and Data Security Conference FC 2011.

[35] Peter Williams, Radu Sion, Miroslava Sotakova, "Private Storage and Making PIR Usable", ACM Transactions on Information and System Security TISSEC 2011.

[36] Bogdan Carbunar, Radu Sion, "Regulatory Compliant Oblivious RAM", Applied Cryptography and Network Security ACNS 2010.

[37] Peter Williams, Radu Sion, Dennis Shasha, "The Blind Stone Tablet: Outsourcing Durability", Network and Distributed System Security Symposium NDSS 2009.

[38] Peter Williams, Radu Sion, Bogdan Carbunar, "Building Castles out of Mud: Practical Access Pattern Privacy and Correctness on Untrusted Storage", ACM Conference on Computer and Communication Security CCS 2008.

[39] Peter Williams, Radu Sion, "Usable Private Information Retrieval", Network and Distributed System Security Symposium NDSS 2008.

[40] Avinash Lakshman and Prashant Malik. Cassandra: a decentralized structured storage system, SIGOPS Oper. Syst. Rev., 44:35–40, April 2010.

[41] The EFF SSL Observatory, online at https://www.eff.org/observatory

[42] EFF Sovereign Keys, online at https://www.eff.org/sovereign-keys

[43] Jonathan M. McCune, Adrian Perrig, and Michael K. Reiter. Seeing is Believing: Using Camera Phones for Human-Verifiable Authentication, International Journal of Security and Networks, Special Issue on Secure Spontaneous Interaction, 4(1/2), pages 43--56, 2009.