Technical Philosophy

Victor Grey
Chief Architect
JLINC Labs

June 15, 2020

JLINC

JLINC is a novel way to assemble methods, systems and techniques to provide confidentiality and to represent fiduciary relationships and accountability for parties sharing data over the global internet. It facilitates control over shared data, provenance of that data, and non-repudiation of data sharing actions.

It makes use of modern standards such as JSON linked data, decentralized identifiers and verified credentials to accomplish these goals.[1]

Antecedents

JLINC builds on previous ideas including capability-based security, especially the work of computer scientist Mark Miller, an influential article entitled *The Strength of Weak Ties* by sociologist Mark Granovetter, the *Augmented Social Network* whitepaper from a group of two dozen professionals in the fields of digital communications, environmental activism, independent media, and socially responsible investment, and *Chain-Link Confidentiality* by professor of law and computer science Woodrow Hartzog.[2]

Granovetter pointed out that a social network consisting only of strong ties is an isolated network. Nobody can be introduced to a new connection since everyone likely already knows everyone else. It is the weak ties – acquaintances or people who have met in passing - who provide bridges between social networks, making introductions for people who otherwise would not have met.

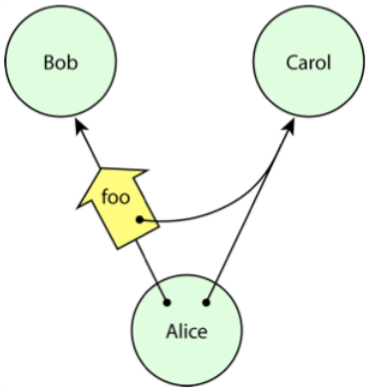
Miller and his co-authors saw this as a way to explain some of the power of capability-based computer systems. Miller says that “In a capability system, *only connectivity begets connectivity*. In a capability system, an object's authority to affect the world outside itself is determined *solely* by what references it holds, since the only way the object can cause an external effect is to send a message via one of these references. Consequently, the mechanics of reference-passing determine how authority can change over time.”[3] He illustrated this with something he called a “Granovetter operator.”

---

[1] https://json-ld.org/
https://w3c.github.io/did-core/
https://www.w3.org/TR/vc-data-model/

[2] https://en.wikipedia.org/wiki/Capability-based_security
http://www.erights.org/elib/capability/ode/overview.html
https://www.aapss.org/fellow/mark-granovetter/
https://www.planetwork.net/asn
http://cyberlaw.stanford.edu/publications/chain-link-confidentiality

[3] http://srl.cs.jhu.edu/pubs/SRL2003-02.pdf



Here, in a diagram from Miller, the yellow arrow foo is a "Granovetter operator" transmitting a capability from Alice to Bob giving him access to Carol. This illustrates connectivity begetting connectivity, as Alice transmits some subset of her ability to connect to Carol over to Bob. Some of the advantages of such a system are that the capability transferred can be as granular as desired, and that no overarching system of identity is required. It is enough that Alice knows both Bob and Carol and no central authority need be invoked.

Another aspect of capability systems is that because capabilities can carry purpose information, they can enable "least authority" systems, in which only the minimum necessary information for a given purpose is released. Again from Miller, "Capability systems provide support for the precise, minimal, and meaningful delegation of authority, which is fundamental to secure operation."

A major influence on the thinking that went into JLINC was the *Augmented Social Network* whitepaper. The ASN embodied and extended the idealism that inspired the originators of the internet decades previously. It talked about interoperability between communities, brokered relationships, and the transitive nature of trust. Nevertheless, it was still decades ahead of its time as far as the realization of its vision was concerned. JLINC makes important steps in that realization.

Finally, there's chain-link confidentiality. Although in fact JLINC had been in development for several years before the inventors became aware of Professor Hartzog's work, his paper *Chain-Link Confidentiality* published in the Georgia Law Review in 2012 lays out the social and legal theory underlying JLINC, which the inventors had correctly intuited. JLINC can be understood as the technical invention that makes the implementation of chain-link confidentiality possible.

Chain-Link Confidentiality states the problem as "Generally, individuals lose control of their personal information once they disclose it on the Internet." And the solution, in essence, is stated as "A chain-link confidentiality regime would contractually link the disclosure of personal information to obligations to protect that information as it is disclosed downstream."

The paper makes the point that contractual confidentiality agreements are a stronger legal framework than the more nebulous notion of "privacy", as they create a fiduciary relationship between the originator of information and the downstream recipients.
And further, that those agreements can be constructed so as to obligate a recipient to assure that any subsequent recipients become parties to the same agreement before any data is transmitted.

This is exactly what the JLINC protocol means by a “Standard Information Sharing Agreement” or SISA. JLINC is the means whereby parties can agree to a SISA in an accountable, auditable, non-repudiable, cryptographically secure fashion, and subsequently have an ongoing exchange of information in the context of the agreed SISA, in the same accountable, auditable, non-repudiable, cryptographically secure fashion.

The Problem

Capability systems were originally conceived as running in the context of a computer operating system whose code enforced the rules, or in a closely networked system all operating under the same coding paradigm. As Lawrence Lessig has stated, “code is law,”[4] meaning that however the code in a computer system is constructed determines what is possible or not possible in that system. He is using “law” in the sense of a law of physics rather than law in the legal sense.

Attempts to organize transactions using so called “smart contracts” have been tried, creating a contract as self-executing code. The problem here is that for practical purposes *all* code contains bugs. The action that the contract executes may, in some unforeseen circumstance, be very much not what the parties to the contract understood would happen. Correction and/or recourse may be problematic since the contract execution is self-evidently in accordance with the contract as coded, as well as irrevocable. One of the first large scale attempts at implementing a “smart contract” resulted in the loss of tens of millions of dollars for exactly this reason.[5]

Attempting to create a capability system between entities over the unregulated public internet poses a problem, because once information is transmitted (i.e. copied) over the internet, the receiving entity is in possession of their copy of it under the control of their own system, and the sender has lost control over the information’s usage by the receiver. Access control at the source of the information is in this sense futile – access can be denied but once it is granted all further control is lost.

Systems have been created to overcome the access problem by restricting the amount or nature of information accessible to that which the parties can agree is absolutely necessary, and no more. The most rigorous of these systems, known as zero-knowledge proofs, are mathematical methods used to verify things without sharing or revealing underlying data. For example, that you have enough money in your bank account to complete a transaction without revealing anything else about your balance.

[4] https://en.wikipedia.org/wiki/Code_and_Other_Laws_of_Cyberspace
[5] https://www.coindesk.com/understanding-dao-hack-journalists

This is not terribly useful however, when two or more parties wish to create an ongoing data stream between themselves containing many types of data (some of it unstructured or of an initially unforeseen nature) but still maintain a fiduciary control over its use. In other words, a confidential conversation.

The Solution

JLINC offers a unique solution. The inventors realized that while a technical solution was possible, it was not possible for that solution to be *purely* technical. It consists of an innovative vision of a capability system, one that functions well on disparate systems running different kinds of code on dissimilar machines in separate organizations.

JLINC is comprised of a key management system, an agreement management system, a communication protocol, and an auditing system, all assembled to work together in a new way to achieve the goal of a capability system that serves to make confidential exchanges between entities secure and provenance-preserving. It introduces a technical means to make fiduciary relationships between parties straightforward to create and audit.

In JLINC, separate computer systems act as agents for entities, either individuals, organizations or other kinds of groups. An agent operates independently to represent the interests of its party to other entities via their agents. Each agent can create any number of entries on behalf of its user in an identity system that creates identifiers that point to the public part of public-private key-pairs.[6] JLINC in its current instantiations uses a method[7] derived from a distributed identity standard called DID,[8] but other suitable systems may arise in the future that would also be compatible with JLINC.

JLINC then provides methods[9] for the parties being represented to select a mutually agreeable contract – a Standard Information Sharing Agreement or SISA – and to mutually cryptographically sign the selected agreement with their respective keys, each retaining a copy and optionally transmitting another copy to an audit service of their choosing. In its current instantiation JLINC uses a standard data representation technique called JSON-LD,[10] although other suitable representation systems may arise in the future that would also be compatible with JLINC.

[6] https://en.wikipedia.org/wiki/Public-key_cryptography
[7] https://did-spec.jlinc.org/
[8] https://w3c.github.io/did-core/
[9] https://protocol.jlinc.org/
[10] https://json-ld.org/

A SISA is a human-readable legally negotiable contract that at a minimum establishes a fiduciary confidentiality agreement between two parties. It requires each party to conform its use of any data transmitted via JLINC from the other to the requirements as stated in the SISA. Such data may include a requirement to conform such use to the preferences of the user as they may change from time to time. These preferences are also transmitted via JLINC as data.

Each data transmission is accompanied by a cryptographic hash of the SISA and signed by the transmitter and receiver, again each retaining a copy and optionally transmitting another copy to an audit service of their choosing. These are called SISA events and provide a non-refutable history of data and preference exchange under the SISA agreement.

SISAs may also be crafted to require a receiving party to only pass on data from the originator to other parties that also use JLINC to agree to the same SISA, thus creating a chain-link confidentiality system.

In the event of a dispute of any kind, the issue may be adjudicated via traditional means, but using the audits of the SISAs and SISA events to establish a cryptographically and mathematically non-refutable history of the facts of the exchanges and agreements between the various parties.

For some use cases a verified credential may be required. JLINC currently uses the Verified Credential[11] standard published by the W3C, but other suitable mechanisms that accomplish the same task may be substituted.

[11] https://www.w3.org/TR/vc-data-model/

Use Cases

Following is a non-exhaustive description of some of the uses made possible with JLINC.

1) News articles and social media postings with verifiable provenance
2) Usage control over personally identifying information (PII)
3) Public pledge signing
4) Contract signing
5) Capabilities for 3rd party sharing with usage and provenance control
6) Notaries Public
7) Verified Attributes
8) Voting and Polling
9) Supply chain verification
10) Fiduciary agent for secure file storage
11) Advertising preferences
12) Group forming and group information sharing
13) Network forming among groups, and information sharing and relationship graphs between groups in a network
14) COVID19 testing and other medical data
15) A microcredit payment system
16) A mutual credit system
17) A sponsorship-based publishing system with verifiable author and sponsorship provenance
18) A networked variation on the sponsorship-based publishing mode
19) A viewer/consumer funded publishing model with verifiable content producer and publisher provenance
20) A wallet-based variation on the sponsorship-based publishing model