Callisto: A Cryptographic Approach To Detect Serial Predators Of Sexual Misconduct

Anjana Rajan Lucy Qin David Archer
Dan Boneh Tancrède Lepoint Mayank Varia

March 29, 2018
Last updated: November 14, 2018

Abstract

Callisto, a non-profit that has created an online sexual assault reporting platform for college campuses, has expanded its work to combat sexual assault and professional sexual coercion in other industries. In our new product, users will be invited to an online *matching escrow* that will detect repeat perpetrators and create pathways to support for victims. Users of this product enter incident details and perpetrator identities into the escrow. This data can only be decrypted by a Legal Options Counselor (a third-party lawyer vetted by Callisto) when at least one other user enters the identity of the same perpetrator. If perpetrator identities match, each user is assigned a Legal Options Counselor, who will connect users to each other (if appropriate) and help each user determine their best path towards justice. User relationships with Legal Options Counselors are structured so that relevant communications benefit from client-counselor privilege. A combination of client-side encryption, encrypted communication channels, oblivious pseudo-random functions, key federation, and Shamir Secret Sharing keep data encrypted so that only Legal Options Counselors gain access to identifying user submitted data when a perpetrator match is identified. In this paper, we present an informal risk management assessment, threat model, and cryptographic solution overview for our new product. A later paper will provide a formal security analysis and mathematical proofs of our cryptographic scheme.

1 The Problem of Sexual Assault and Harassment

An estimated 20% of women, 7% of men, and 24% of transgender and gender non-conforming students are sexually assaulted during their college careers. Less than 10% of survivors of such assault report those incidents to administrators, local police, campus security, or other authorities. Those who choose to report do so an average of 11 months after their assault, making it hard to conduct an effective investigation. Those investigations are not challenging because perpetrators are unknown – in fact, 85% of college survivors know their assailant – but rather because investigators are not sure whether to believe that an assault actually took place. Only 6% of assaults reported to the police end with the assailant spending a single day in prison, meaning that over 99% of those perpetrators will not face serious consequences for their actions. Thus, there is at present no effective deterrent to sexual assault in the United States [4, 3].

Two facts suggest a solution direction that motivates Callisto's new capability. First, an estimated 90% of college sexual assaults are committed by repeat perpetrators. These serial perpetrators assault an average of 6 times before they graduate from college. Unfortunately, with such a low reporting rate, it is fairly unlikely that even serial perpetrators will be reported, much less reported more than once. As a result, investigators often have no knowledge of a pattern of behavior of the accused when trying to make a fair judgment on a case. Without clear evidence (which is hard to gather if a report is made months after an assault) or a pattern of behavior, authorities are often hesitant to assume liability for taking action against an accused perpetrator. It is far more common for colleges to be sued for expelling an accused sexual assailant than for neglecting the safety and privacy rights of victims.

Second, in a survey conducted by Callisto of over 200 college sexual assault survivors, a clear pattern emerged – for most victims, reporting an incident of sexual assault was not worth it, unless they knew they were not the only victim of that assailant. Learning of another victim of the same assailant dramatically increased victims' willingness to report, as well as their perceived likelihood of being believed if they reported. We might create a sea change if victims could learn that they are "not the only one". However, in our current environment, the only way they can learn of other victims is through the "whisper network", or if their perpetrator is identified in the press or on social media.

As the #MeToo movement has manifested, it has become clear that sexual assault and harassment, especially in the workplace, is prevalent across many industries. Incentives have shifted so that employers and professional sectors are now facing the same kind of pressure that colleges faced 4 years ago: pressure to take tangible action to address sexual misconduct. However, employers face many of the same issues as college investigators: delayed reports with little evidence other than the testimony of the victim, and with no good way to learn whether the accused exhibited a pattern of behavior. Victims in the workplace face the same stark reality as college survivors – that it is often not worth it to report unless you know that you are not the only one. The Equal Employment Opportunity Commission's 2016 report on workplace harassment found that almost one third of the approximately 90,000 complaints received by the EEOC in fiscal year 2015 included an allegation of workplace harassment [8]. Roughly three out of four employees who experienced harassment never talked about it with a supervisor, manager or union representative. Employees choose not to report or file a complaint because they fear disbelief, inaction by management, blame, or social or professional retaliation. While everyone can agree that the current equilibrium does not work, it remains in the best interest of both victims and authorities to continue the culture of silence. If nothing changes, once the #MeToo movement fades, victims will continue to not report, authorities will resume not taking action, and serial perpetrators will continue to assault and harass more victims.

Unfortunately, under-reporting by victims and associated non-accountability of perpetrators is not the only problem to be solved. As much as we believe that efforts to stop sexual assault and harassment would benefit from full disclosure and transparency, that same ideal works against victims. A victim's identity, details of incidents, and identities of perpetrators are all highly sensitive information. In the wrong hands, that information can be used to cause serious harm. Worse, it can be used to inhibit the reporting and follow-up so necessary to helping victims find justice. Perpetrators would certainly use the knowledge that they might be reported to intimidate, threaten, or take legal action against victims. More importantly, society often uses such information to damage victim or perpetrator reputations or wellbeing. Finally, and most importantly, each victim's right to choose their path to justice is paramount. Victims need a way to discover the paths open to them and find support on those paths *while retaining their privacy and personal security.*

2 Solution Overview

Callisto approaches the problem of hesitance in reporting assault by using the mathematical tool of *game theory*: a way of modeling situations of conflict among parties [2]. In game theory terms, there is a first-mover disadvantage with high consequences for a victim when accusing a perpetrator. That disadvantage comes from the disclosure and resulting exposure of the victim, opening the victim up to consequences (countermoves in game theory) of retaliation, disbelief by authorities, reputation damage, and stigma. Callisto's solution leverages the two key facts described in the previous section to eliminate first-mover disadvantage: we enable the likely multiple victims of a perpetrator to know they are not alone and create a path for them to act together. This approach disincentivizes retaliatory countermoves by perpetrators while supporting combined action by victims that reduces disbelief by authorities and likelihood of reputation damage.

Callisto approaches the problem of protecting victim (and perpetrator) privacy through comprehensive use of privacy-preserving encryption technologies and user authentication practices. Personal information of users, their accounts of incidents, and the identities of perpetrators they provide are protected by encryption from before that data leaves the user's personal computer. All such data stays encrypted until decryption on the personal workstation of a Callisto Legal Options Counselor. In addition, even Counselors cannot see incident or perpetrator identity information *unless more than one user has identified the same perpetrator*. Access to user accounts are protected by multi-factor authentication, strong password requirements, and e-mail verification. User activity on the platform cannot be linked to identifying user account information.

Callisto's new offering is available to users by invitation only. Invited users will receive an e-mail invitation to activate accounts on Callisto's system. They will then verify their e-mail address. Once verified, users may submit incident entries, modify them, or delete them at will. Each entry includes the identity of the accused perpetrator, which may be in one or more of several forms: an email address, a cell phone number, or a social media URL. Upon submission, encrypted data is categorized in two sets: Assignment Data and Entry Data. This is discussed in further detail in section 4.3. When matches are identified, Assignment Data is triaged by the Director of Legal Options Counseling (DLOC), a Callisto attorney, who manages the network of third-party Legal Options Counselors (LOCs). The DLOC assigns each matched user their own LOC who will review Entry data and reach out to that user individually to help them find their desired pathway to justice.

3 Legal Framework

3.1 Categorization of Data

Callisto's platform and services are designed to provide maximum protection of Entry Data, and to empower the victim with the choice of whether to share such information, with whom, and when. Entry Data is comprised of the following:

Identifying Data

- Perpetrator name
- Perpetrator unique identifiers, including social media account information
- Victim preferred name
- Victim preferred phone number
- Victim preferred contact email
- Victim preferred contact method (phone, email, text, chat)

Non-Identifying Data

- State of current residence of Matched User
- Categorization of sexual misconduct
- Industry of perpetrator

The non-identifying data is a subset of Entry Data that enables the assignment of a Matched User to the best suited LOC without directly or indirectly identifying either the user or perpetrator. It is also referred to as Assignment Data.

3.2 Legal Constraints

Callisto has designed the platform and services so that, to the extent possible, Entry Data is "not discoverable," meaning that it is protected from litigation or investigation discovery requests (including subpoenas served to Callisto). Prior to data relating to victims and perpetrators being shared with the Legal Options Counselors, processes are put in place to ensure that all victims agree to sharing the data they submitted in their entry with their assigned Legal Options Counselors. Information is generally not discoverable when (i) it is encrypted and no-one has the ability to decrypt it, or (ii) someone has the ability to decrypt it, but the information is otherwise protected by a privilege, such as the attorney-client privilege protection. The platform is designed so that Callisto staff does not have at any point in time (before and/or after a match) any access to any identifying Entry Data. In order for identifying Entry Data to be protected from discovery via encryption, no Callisto employee can have the capability to decrypt it. If Callisto receives a subpoena for encrypted data, and a Callisto employee has access to the key to decrypt such data, then Callisto will be obligated to (i) compel its employee to use the key and decrypt the data and (ii) produce the decrypted data in response to the subpoena. Therefore, no Callisto employee has access to any decryption key at any point in the process (including after a match has occurred) that would permit access to identifying Entry Data. Callisto designates a third party to receive the decryption key and use it under a certain set of circumstances (e.g. Legal Options Counselor's key can unlock Entry Data upon a match of two victims with a single perpetrator). Although these Legal Options Counselors could receive a subpoena for the encrypted data, the data is protected by a specific privilege, i.e. a special legal protection of the confidentiality of the data that makes it not discoverable. Since the Legal Options Counselor is a third party attorney, and the purpose of receiving access to the Entry Data is to deliver legal advice to the victim, then the Entry Data will be considered an attorney client communication and will be protected by the attorney-client privilege. Therefore, Callisto has required that any third party who has a decryption key that can unlock Entry Data at any point in the process 1) is an attorney and 2) receives the Entry Data as a communication from the victim the purpose of which is to seek legal advice.

3.3 User Pin

Once a user creates an account, the user provides an account email (preferred email) as part of their login credentials, as well as a 4-digit pin generated by the user. Callisto (1) hashes the preferred email together with the 4-digit pin, and (2) public key encrypts the preferred email with a key stored on the communication server. Public-key encryption with a key stored on the communication server enables Callisto to send password reset emails as well as general product update emails to account holders. Hashing the preferred email together with the user generated 4-digit pin adds a degree of randomness which prevents linking a user's preferred email address to their unique account ID (such account ID is linked to all activities of the user on Callisto, including the creation of an entry and the entry's match to another user account ID and to a perpetrator ID). User activity on Callisto (creation of an entry, matching with another victim, being assigned a Legal Options Counselor) is therefore not tied to a user's email address or any other identifying account information. If Callisto receives a subpoena requesting activity information about a specific individual, Callisto would not be able to provide any relevant information.

4 Callisto's Privacy Risk Management Framework

Callisto takes the responsibility of protecting user privacy seriously. Threats to privacy of information in a computer system come from a variety of adversaries: *external* adversaries that may break into a system in a variety of ways, from brute force cyber exploits to compromising user credentials; *insider* adversaries who may have authorized access to a system and use those rights inappropriately; and *imposter* adversaries who pretend to be genuine system users in order to see what can be learned.

Our early security stance assessment used a lightweight form of the National Institute of Standards and Technology Risk Management Framework (NIST RMF) assessment [9]. In addition, our design was reviewed in depth by a security assessment team, and all discovered issues were addressed before platform release. In addition, we will continually self-test our systems for security vulnerabilities after platform release, and continuously apply patches and updates. Here, we summarize our initial NIST RMF assessment by providing an overview of the information assets we aim to protect for users and perpetrators, and our approaches for protecting those assets. We describe these assets roughly in the order they appear in the user experience flow of our product.

4.1 Sensitive Information Assets Held by Callisto

Invited Users. Callisto provides access to the platform and counseling services (as applicable) through white-listing email accounts. *Invited Users* are members who are email-invited to have access to Callisto. The information Callisto learns about Invited Users includes:

- *Invitation email address* – email addresses are sensitive: they may include names, and may be correlated with other external information held by a potential adversary;
- *Organization name (as applicable)* – organization names are sensitive: the relationship between an invited user and an organization may not be public knowledge. In addition, the customer relationship with Callisto may not be public knowledge.

Activated Users. An *Activated User* is an Invited User who visits the Callisto site and registers an account. The information Callisto learns about Activated Users includes:

- *Preferred email address* – the hash of a preferred email address concatenated with a user-submitted 4-digit pin;
- *Access to Callisto* is sensitive, so we protect user access in multiple ways: we protect each Activated User's passphrase and use additional authentication factors to verify user access rights.

Escrowed Users. An *Escrowed User* is an Activated User who has submitted one or more records of sexual assault or harassment. If an adversary discovers that a user is escrowed, or exfiltrates those records, it can cause significant damage to the user and to the perpetrator. This information includes Entry Data as described in Section 3.1.

Assigned Users. An *Assigned User* is an Escrowed User who has a perpetrator match with another user. Once such a match has been discovered, the Director of Legal Options Counseling has access to Assignment Data as described in Section 3.1. Recall that Assignment Data does not directly or indirectly identify the user or perpetrator, but is sufficient to assign a Legal Options Counselor to the user.

Matched Users. A *Matched User* is a user who is assigned a Legal Options Counselor. LOCs help survivors navigate their options, whether legal or not, weigh benefits and risks of action, and explore whether they want to come forward, and in what manner. The Legal Options Counselor has access to Entry Data. Users and LOCs engage in direct conversations protected by attorney-client privilege.

Callisto Legal Options Counselor Credentials. The Director of Legal Options Counseling and the network of Legal Options Counselors have sensitive access credentials that are initially generated by Callisto.

We protect:

- *Account passwords*, even though a login alone does not allow access to sensitive user or incident information described above;
- *Secret access keys* that the DLOC or LOCs use to access encrypted information about users.

Callisto Information Technology Support Personnel. Callisto IT support personnel have no access to the encrypted information about users and incidents described above. However, an adversary that compromises the credentials of any IT personnel may be able to corrupt or destroy that information, and thus harm our users and our business. *Access to Callisto systems* by IT personnel is secured in the same way as access by Users and Counselors.

Callisto Data Analysts. Callisto uses statistics about incidents and progression through the legal options counseling workflow in order to measure user impact. For this reason, Data Analysts have access to options counseling workflow state statistics, but no access or cryptographic keys to user information, perpetrator identities, or incident records. However, even this statistical information may be sensitive, so we secure access by our analysts in the same multi-factor way that we use for users, Counselors, and IT personnel.

4.2 Technical approaches for securing sensitive information at Callisto

Table 1 below describes, for each asset type above, how we protect the asset type, as well as who has access under what conditions.

5 Cryptographic Design

5.1 Roles in our Cryptographic Design

Data submission involves interaction between several parties: the user's browser, the Callisto application server, and the Callisto key server. Data is stored within a relational database on the Callisto application server. The key server serves two roles: it stores a predetermined key whose purpose is explained below, and it authenticates Callisto users during the login process.

Information Asset Type	How Protected	Decryptable by
Invitation e-mail address	Public key cryptosystem	Callisto
Preferred e-mail address	Bcrypt hash of the SHA-256 hash of the e-mail address concatenated with the 4-digit pin	N/A
User authentication	Multi-factor authentication service	N/A
User passphrase	Randomly generated string of 6 words	N/A
Assignment data	see Crypto Design, below	DLOC (only after a match)
Entry data	see Crypto Design, below	LOC (only after a match)
DLOC and LOCs secret keys	Hardware authentication device	N/A

Table 1: Callisto Sensitive Information Assets

5.2 Cryptographic Components

Our system is designed to ensure that a compromise of one of the Callisto servers, the application server or the key server, *reveals no information about Assignment and Entry Data.* To achieve this goal the Callisto system uses the following cryptographic components:

- Shamir Secret Sharing [10]: Let s be a secret key. Shamir Secret Sharing is a technique that lets us split s into many shares $s_1, s_2, \ldots, s_n$ so that (1) a single share s_i reveals nothing about s, and (2) when two shares become public, anyone can reconstruct the secret s. Briefly, to create shares of s we generate a random line in a plane of possible secret shares whose y-intercept is the secret s. The shares of s are points on this line. A single point reveals nothing about the line, but two points reveal the line and thus enable computing its y-intercept.

- Oblivious pseudo-random functions (OPRFs). An OPRF uses a secret key k_s to map a value x to a pseudorandom value $\hat{x}$ [7]. This secret key k_s is stored on the Callisto key server. A client who has an input x can interact with the Callisto key server to obtain $\hat{x}$. The "oblivious" property refers to the fact that in this process, the key server learns nothing about x, yet the client learns $\hat{x}$. We stress that this process is deterministic: evaluating the OPRF at the point x (using the key k_s) always results in the same pseudorandom value $\hat{x}$.

- Symmetric encryption. For a given secret key k and message m we will use $c \leftarrow E(k, m)$ to denote the encryption of m using key k. We will use $D(k, c)$ to denote the decryption process. Callisto uses `libsodium`'s default implementation for symmetric encryption [5].

- Public key encryption. We will use $c \leftarrow \mathcal{E}(pk, m)$ to denote the encryption of m using public key pk, and $\mathcal{D}(sk, c)$ to denote the decryption of c using the corresponding secret key sk. Callisto uses `libsodium` for public key cryptography [5].

5.3 The Data Submission and Protection Process

The Callisto key server is initialized to hold the OPRF secret key k_s. The database server holds no secrets. To simplify our description here, we describe a single Callisto Legal Options Counselor. That LOC generates a key pair *pk* and *sk* for a public-key encryption scheme and makes the public key *pk* available to the public.

With this setup complete, we informally describe the cryptographic portions of workflows for (1) entry submission, (2) entry encryption, (3) entry editing, (4) perpetrator matching, and (5) revealing information to the DLOC and LOC in our system at a high level. Full details, along with a cryptographic security model, will be available in a forthcoming paper.

Submitting an Entry. The user's computer (the client) collects the Entry Data from the user and formats it into a serialized structure. As mentioned in the legal framework, the Assignment Data is derived as a subset of the Entry Data and contains only the fields necessary for a DLOC for assigning a user to a LOC.

Next, the user authenticates to the key server, and once authenticated, the user's client interacts with the oblivious pseudo-random function (OPRF) system on the key server to transform the *low-entropy* perpetrator's ID P into a *pseudorandom* value $\hat{P}$ with sufficient entropy for use in our secret sharing approach. During this step, the key server learns the authentication token, but learns nothing about P from the user's client. Only the user's client learns $\hat{P}$.

The client then creates a secret share. It uses $\hat{P}$ to derive three 32-byte pseudorandom quantities (a, k, π) using the key derivation function in `libsodium`. The first two quantities define a line equation $Y = aX + k$ whose y-intercept is k. The client evaluates this line equation at the point $X = U$ to obtain $s = aU + k$, where U is the hash of the user's unique id. The pair (U, s) is one share of a Shamir Secret Sharing Scheme for the secret k. All arithmetic operations are performed modulo a prime number; Callisto uses the prime $p = 2^{256} + 297$.

Encrypting an Entry. The client encrypts Entry Data using a fresh, random entry data key k_e to obtain an encrypted entry $\texttt{eEntry} \leftarrow E(k_e, \texttt{EntryData})$. It then encrypts k_e twice, once using the key k generated above from $\hat{P}$, and once using a user key k_U which is discussed further below:

$$c_e \leftarrow E(k,\ k_e), \qquad c_U \leftarrow E(k_U,\ k_e).$$

All these symmetric encryptions are done using authenticated encryption with associated data (AEAD) where π and a pre-defined string are used as the associated data. The client then performs one more encryption, encrypting (U, s, c_e) under the Legal Options Counselor's *public key* pk_{LOC} to obtain a doubly-encrypted ciphertext:

$$c \leftarrow \mathcal{E}\big(pk,\ (U, s, c_e)\big).$$

A new key, k_a, is randomly generated and used to encrypt Assignment Data, creating the following: $\texttt{eAssign} \leftarrow E(k_a, \texttt{AssignmentData})$. Using the same k, we obtain $c_a \leftarrow E(k, k_a)$. Once again, the symmetric encryptions are done using authenticated encryption with associated data (AEAD) where π and pre-defined strings (different as above) are used as the associated data. The public key of the DOC is used to encrypt

$$c_{assign} \leftarrow \mathcal{E}\big(pk_{DLOC},\ (U, s, c_a)\big).$$

The client authenticates to the Callisto database server and sends it the tuple

$$(\pi,\ c,\ c_{assign},\ c_U,\ \texttt{eEntry},\ \texttt{eAssign}). \tag{1}$$

The database server stores this tuple and sends an acknowledgement to the user's browser. On its own, this tuple (1) reveals nothing about the Entry Data. Not even the Legal Options Counselor can decrypt eEntry, since it is not possible for them to construct the encryption key k using c and c_a, respectively, for the DLOC and LOC. Moreover, if a user submits two records about the same P, this second record will result in the same share (U, s) as the first record, and thus nothing new is revealed about P. Finally, nothing about the submission process reveals anything to the user's browser about other entries or perpetrator identities.

Editing an Entry. The user key k_U makes it possible for the user to update the record after the initial submission, if needed. The key k_U is generated on the user's client at initial submission time and the user is asked to write down this key as a string of random words. When the user needs to update the submission, the user types in this key and the user's client uses it to update the encrypted entry eEntry by first retrieving $k_e \leftarrow \mathcal{D}(k_U, c_U)$ to produce $\texttt{eEntry}' \leftarrow E(k_e, \texttt{EntryData}')$.

If the perpetrator identity, P, is updated with P', this new value is used to perform an OPRF with the key server, creating $\hat{P}'$ and thus π', k', U', and s'. This produces:

$$c'_e \leftarrow E(k', k_e) \qquad c' \leftarrow \mathcal{E}(pk, (U', s', c'_e))$$

If AssignmentData has been updated, a new random key, k'_a, is generated and used to produce $\texttt{eAssign}'$. If P was updated,

$$c'_a \leftarrow E(k', k'_a) \qquad c'_{assign} \leftarrow \mathcal{E}(pk_{DLOC}, (U', s', c'_a))$$

Otherwise,

$$c'_a \leftarrow E(k, k'_a) \qquad c'_{assign} \leftarrow \mathcal{E}(pk_{DLOC}, (U, s, c'_a))$$

For each item in (1) that has been updated, the original is replaced. For example, when both the perpetrator identity and some field in AssignmentData has been updated, the following tuple is stored in the database.

$$(\pi',\ c',\ c'_{assign},\ c_U,\ \texttt{eEntry}',\ \texttt{eAssign}'). \tag{2}$$

Matching Entries. The Callisto database server periodically performs an offline match search. If it finds two entries with the same π component, it identifies these records as a match, because they share a common perpetrator. It then notifies the DLOC about the match. Then, the DLOC assigns each matched user to their own LOC via the Callisto platform.

Note that matching is done without the database server having access to perpetrator identities or entries in unencrypted form. Thus no adversary capable of penetrating the database server can learn anything about perpetrator identities or incidents from the matching process.

Decrypting Entries. When the database server identifies a match, the DLOC and LOC can then obtain the (U, s) values for all matched entries using their respective keys, sk_{DLOC} and sk:

$$(U, s, c_a) \leftarrow \mathcal{D}(sk_{DLOC}, c_{assign}) \qquad (U, s, c_e) \leftarrow \mathcal{D}(sk, c)$$

.

Once at least two shares are decrypted, they can be used to derive the slope, a and the original k. Using this k, the DLOC and LOC can decrypt the following, respectively.

$$k_e \leftarrow D(k, c_e) \qquad k_a \leftarrow D(k, c_a)$$

For each c_e and c_a that can be decrypted with a valid k, the resulting k_e or k_a values can be used to then decrypt `eEntry` and `eAssign`. The decryption algorithm is described in more detail in the next section.

6 Additional Details

This paper is too brief for an exhaustive description of our cryptographic approach to protecting incidents, perpetrators, and connections to the Escrowed Users reporting them. However, we include here some additional details that may be of interest to readers.

Identifying Perpetrators. Escrowed Users may identify Perpetrators using one or more diverse credentials such as social media URLs, phone numbers, or email addresses. In our system, we insist on the use of such (relatively) unambiguous identifiers. To allow for this diversity of credentials, π and s are not scalars. Instead, they are *vectors* of values $\vec{\pi}$ and $\vec{s}$, where each component in these vectors corresponds to a particular predetermined type of identifying credential. We say that two records $(\vec{\pi}_1, \ldots)$ and $(\vec{\pi}_2, \ldots)$ are a match if the vectors $\vec{\pi}_1$ and $\vec{\pi}_2$ match on at least one component. Moreover, the Callisto Legal Options Counselor workstation can fill in additional components in the $\vec{\pi}$ and $\vec{s}$ vectors for an incident once such a match is determined, because they have access to the necessary resources. Thus Callisto *propagates* perpetrator identities, using the human judgment of our Counselor, to achieve more complete identity credential vectors for perpetrators.

Privacy Roots of Trust in Callisto. Every system has one or more *roots of trust*: one or more components that are assumed secure in certain ways. Briefly, our system assumes the following privacy protecting roots of trust:

- The user's browser (and computer) are one root of trust in that we assume no adversary has compromised that component with the intent of observing interactions with our system. In other words, protecting against system adversaries with vantage point on the user's computer is *out of scope* for our system. Users are responsible for adequately protecting the passphrase they use to log in, as well as the devices and accounts they use for multi-factor authentication.

- Above, we described a system using a single key server. The OPRF key is a highly sensitive secret in our system. If this key is exposed to an adversary, then that adversary can unmask records in the database by performing an exhaustive search over potential perpetrator identities. In addition, if this key is lost, then matching post-loss perpetrator identities to pre-loss identities is impossible. To further protect the OPRF key, our system uses two servers, each of which keep a single cryptographic share of the key, but not the whole key. Thus key theft requires compromise of two distinct servers with different administrators, and possibly running different operating systems. This *split server* is another root of trust of our system. One of these servers is a dedicated, highly protected physical server. The other will be a virtualized server that is hosted on a separate cloud provider. To prevent loss of the OPRF key, both servers are backed up in an encrypted backing store. To further thwart dictionary attacks, the key servers will perform rate limiting.

- The Legal Options Counselor's workstation contains a password vault used to hold the Counselor's secret key that enables decryption of user profiles, as well as incident records and perpetrator identities (these latter two only after a match, as described above). This vault is in turn protected by a passphrase known only to the Counselor. Such passphrases are a *partial* root of trust for our system. We may increase security in this area by storing cryptographic shares of the Counselor's private key in a distributed fashion, and performing decryptions without bringing those key shares together in the clear, ever.

- The database server is *not* a privacy root of trust for our system, because it holds no secret keys, and because all sensitive information held there is encrypted with keys held on other components in our system.

Managing Multiple Legal Options Counselor Keys. Above we describe a single Legal Options Counselor but in reality there may be multiple counselors who need to access the same information. Each Legal Options Counselor has a distinct public and private key pair. For each Legal Options Counselor, there will be a copy of c' encrypted under that public key. To add a new LOC, Bob, a first LOC, Alice, would need to decrypt an encrypted share and re-encrypt as follows.

$$c_{Alice} \leftarrow \mathcal{E}\big(pk_{Alice},\ (U, s, c')\big)$$
$$(U, s, c') \leftarrow \mathcal{D}(sk_{Alice},\ c'_{Alice})$$

Once this share is decrypted, Bob can now encrypt its contents with his public key.

$$c_{Bob} \leftarrow \mathcal{E}\big(pk_{Bob},\ (U, s, c')\big)$$

Bob now has access to the shares and can decrypt all matched data. In order to remove Bob from the system, each ciphertext c_{Bob} encrypted under his key is removed from the database.

Decryption Algorithm. Since the code for generating the (U, s) values occurs on the client side, a malicious adversary could alter the (U, s) values of their share, producing combinations that cannot be used to derive a valid key, k. To mitigate this threat, the decryption algorithm is provided a vector of matched *shares* containing (U, s) values as well as ciphertexts to be decrypted. The algorithm first tries to interpolate the first two (U, s) values. If a valid key, k, is found, it is used to decrypt as many remaining c_e or c_a values as possible. Otherwise, it tries to interpolate and find a valid k with all remaining shares. If no valid key is found, an error message is produced for the current share and the algorithm moves onto repeat this process for the next share. This process is described below.

Algorithm 1 Interpolation and Decryption Algorithm

```
procedure DECRYPT($shares$)
    while $shares$.length $> 0$ do
        $flag \leftarrow false$
        $s \leftarrow shares$.pop()
        for each $t$ in $s$ do
            $k \leftarrow$ interpolateShares($s, t$)
            if symmetricDecrypt($k, s$) then
                $flag \leftarrow$ true
                for each $u$ in $s$ do
                    if symmetricDecrypt($k, u$) then
                        $shares$.remove($u$)
                break;
        if !$flag$ then
            unableToDecrypt($s$)    ▷ Creates an error message corresponding to the id of s
```

The *symmetricDecrypt* function mentioned decrypts either c_e or c_a using k. It then uses the resulting key to decrypt either `eEntry` or `eAssign`.

7 Demo Application

A demo application is made available to convey our methods for client-side encryption and Shamir Secret Sharing in a user-friendly format. It can be found at

https://cryptography.projectcallisto.org.

Unlike the Callisto application, the demo is set up as a single server with the browser simulating interactions between the client and the various servers. The demo application does not reflect the full cryptographic functionality of the product. The table below indicates specific functionalities where the encryption strategies are different in the demo application versus the full product.

Functionality	Full Product	Demo
PerpID inputs	Vector of IDs	Single ID
PerpID randomization	OPRF_{k_s} with key servers	$\text{SHA512}(\text{PerpID} \| k_{demo})$
Symmetric Encryption	AEAD	AE
Data storage	Callisto database	Browser
Matching	Across vectors	Equality between IDs

Table 2: Implementation Differences

k_{demo} is a pre-selected string with no correlation to k_s

8 Subsequent Work

In the time since release of our original white paper describing the new Callisto platform, two relevant subsequent works have appeared in the literature. Each of these bears some resemblance to our platform. In this section we briefly comment on both of these works, comparing and contrasting with Callisto where applicable.

SATE [1]. In October 2018, Arun *et al.* proposed *SATE*, a cryptographic solution for an allegation escrow system. Their system uses dynamic multi-party computation and verifiable pseudo-random functions to assure security of sensitive information. In contrast, our platform splits trust among two independently managed and located key servers, each of which hold cryptographic shares of the master secret key we use. In addition, our database computation servers hold no key material or decrypted data – only client browsers of LOCs can decrypt Entry Data, and only client browsers of DLOCs can decrypt Assignment Data.

SATE relies on a collective of external parties that together compute required randomness in a verifiable way and together securely perform matching and decryption. If the majority of these parties act honestly, confidentiality and functionality are preserved. Unfortunately, human trust does not extend easily to the kind of collectives upon which SATE relies. One reason is that in such a distributed model, accountability is difficult to reason about or assign, while in general humans tend to conflate trust with accountability. Another reason is that forming such collectives is tricky in practice because of the nuanced difference between incentivizing participation and gaining undue influence over that participation (and thus raising questions about security). In contrast, the Callisto choice of a single organization is intentional: users need extend trust to no-one initially, and then only to legal representatives (LOCs) contracted by a single organization (Callisto) that is independent of any entity where incidents might occur. Risk of coercion of Callisto is minimal: because of its funding model and non-profit status, Callisto owes allegiance to no-one that might be a reported perpetrator.

SATE requires user pre-registration with a trusted authority in order to link real-world identities to user accounts (and to user cryptographic keys). While authenticating users in this way is an interesting potential mitigation against malicious users, we felt it to be unreasonable in our application domain for two reasons. First, pre-registration allows the trusted authority to know the identity of users registering, and thus to know who is likely using the SATE service. Possesssion of this knowledge turns the trusted authority into a security threat. Second, pre-registration requires interaction by each user with a third party, which complicates the user experience in ways we felt unnecessary. In contrast, Callisto aims to keep the user experience as simple as possible. We address the challenge of identifying potential malicious users with a combination of controlled access by invitation only, and by human judgment when reviewing matched records.

SATE allows survivors to choose the threshold of matches required for their data to become accessible to the lawyer – a nice feature that offers flexibility to users. In contrast, we selected a fixed threshold of 2 for Callisto, though our solution can be trivially extended to support any value $n \geq 2$ using n-way secret sharing instead of 2-way secret sharing. Our choice was based on usability considerations, especially in light of the emotional distress we recognize users may feel when reporting incidents: asking a user to choose how many matches are required seems somewhat unreasonable without some well-understood criteria by which to choose that value.

SATE is proven secure in the UC model - nicely done! Our security proof will appear in a future paper.

EConfidante [6] puts forward the idea that access to sensitive data can be effectively managed by employing a cluster of Intel Software Guard Extensions (SGX) enclaves that share a secret decryption key unknown outside enclave security boundaries. This system also uses a blockchain approach to permanently store such sensitive data. Unfortunately, EConfidante's design raises several concerns, a few of which we list here. For example, decisions about when to report incidents to authorities appear to be fully robotic, lacking any concept of human intelligent decisions and sensitive reasoning about each victim's desired pathway to justice. Similarly, there appears to be no provision for human judgment in deciding whether claims might be false or spurious. As another example, the choice of a blockchain as a shared database prevents records from being deleted, ignoring the user's "right to be forgotten", or to have an incident report deleted entirely. Another seemingly important issue is that if the EConfidante enclave codebase changes (and of course it will – all software has defects that must be repaired), then the entire system must be brought down and started up again. Because the leader self-generates a new key with which all data is encrypted, it appears (if our understanding is correct) that forward secrecy becomes a problem in this situation: scanners cannot scan ledger entries written with different versions of the running enclave code, so past reports become unusable for matching.

9 Conclusion and Next Steps

Callisto envisions a world where sexual assault and harassment are rare and survivors are supported in their pursuit of justice. The reporting experience should be empowering for survivors and should rebuild their sense of agency. Authorities should have the data they need to prevent assault and stop serial perpetrators.

A thoughtful cryptographic design is essential in achieving this mission. In order to create a safe space for survivors of sexual assault to come forward with their most vulnerable secret, it requires organizations like Callisto to build a technical solution that earns their trust.

As we expand to serve and empower more users, we realize that our threat models will become more sophisticated and complex. Therefore, as we expand from college campuses into industry, our solution has evolved to protect against those risks.

10 Acknowledgements

We would like to acknowledge the Software & Application Innovation Lab (SAIL) within the Hariri Institute for Computing at Boston University, particularly Frederick Jansen, Andrei Lapets, and Kinan Bab, for their work in the software implementation of this cryptographic approach. Lucy Qin and Mayank Varia are partially supported by NSF grants #1718135 and #1414119. Dan Boneh is supported by NSF, DARPA, ONR, the Simons Foundation, and a Google faculty fellowship. Opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of these organizations.

References

[1] Venkat Arun, Aniket Kate, Deepak Garg, Peter Druschel, and Bobby Bhattacharjee. SATE: Robust and private allegation escrows. https://arxiv.org/abs/1810.10123, 2018.

[2] Ian Ayres and Cait Unkovic. Information escrows. *Michigan Law Review*, 111:145–196, 2012.

[3] Marcus Berzofsky Bonnie Shook-Sa Christopher Krebs, Christine Lindquist and Kimberly Peterson. Campus Climate Survey Validation Study Final Technical Report. https://www.bjs.gov/content/pub/pdf/ccsvsftr.pdf, 2016. [Accessed: March 26, 2018].

[4] Susan Chibnall Reanne Townsend-Hyunshik Lee Carol Bruce David Cantor, Bonnie Fisher and Gail Thomas. Report on the AAU Campus Climate Survey on Sexual Assault and Sexual Misconduct . https://www.aau.edu/sites/default/files/AAU-Files/Key-Issues/Campus-Safety/AAU-Campus-Climate-Survey-FINAL-10-20-17.pdf, 2017. [Accessed: March 27, 2018].

[5] Frank Denis. libsodium. https://www.npmjs.com/package/libsodium. [Accessed: November 11, 2018].

[6] Danny Harnik, Paula Ta-Shma, and Eliad Tsfadia. It takes two to #metoo - using enclaves to build autonomous trusted systems. *CoRR*, abs/1808.02708, 2018.

[7] Katrina Ray Ryan Speers-Brian Vohaska Jonathan Burns, Daniel Moore. Ec-oprf: Oblivious pseudorandom functions using elliptic curves. IACR Cryptology ePrint Archive, 2017.

[8] Victoria Lipnic. EEOC Select Task Force on the Study of Harassment in the Workplace. https://www.eeoc.gov/eeoc/task_force/harassment/. [Accessed: March 16, 2018].

[9] NIST. Risk Management Framework. https://csrc.nist.gov/projects/risk-management/risk-management-framework-(RMF)-Overview/. [Accessed: March 21, 2018].

[10] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

A History

- **March 29. 2018.** Initial version of the document.
- **November 12, 2018.** Key changes include the addition of the legal framework and the distinction in the encryption workflow for data accessible to the Director of Legal Options Counseling (DLOC) vs. Legal Options Counselors (LOC), and the subsequent work section for [1, 6].