Balancing Societal Security and Individual Privacy: Accountable Escrow System

Jia Liu
School of Computer Science
University of Birmingham, UK
liujw@cs.bham.ac.uk

Mark D. Ryan
School of Computer Science
University of Birmingham, UK
m.d.ryan@cs.bham.ac.uk

Liqun Chen
HP Laboratories, Bristol, UK
liqun.chen@hp.com

***Abstract*—Privacy is a core human need, but society sometimes has the requirement to do targeted, proportionate investigations in order to provide security. To reconcile individual privacy and societal security, we explore whether we can have surveillance in a form that is verifiably accountable to citizens. This means that citizens get verifiable proofs of the quantity and nature of the surveillance that actually takes place. In our scheme, governments are held accountable for the extent to which they exercise their surveillance power, and political parties can pledge in election campaigns their intention about reducing (or increasing) this figure.**

We propose a general idea of *accountable escrow* to reconciling and balancing the requirements of individual privacy and societal security. We design a balanced crypto system for asynchronous communication (e.g., email). We propose a novel method for escrowing the decryption capability in public-key cryptography. A government can decrypt it in order to conduct targeted surveillance, but doing so necessarily puts records in a public log against which the government is held accountable.

1 Introduction

The world learned that the USA and other Western nations are amassing data about the minutiae of our daily lives on an unprecedented scale, when the former intelligence worker Edward Snowden began an ongoing series of revelations in June 2013. The data includes all forms of electronic communications between people, as well as web accesses, and financial and transport data, and the physical movements of people collected through mobile phone location tracking.

Much outrage has justifiably been expressed by academics [6], [8], [9], [1], politicians [2], journalists [11], and, somewhat hypocritically, by the very companies that enabled it to happen [10]. Nevertheless, the purpose of this mass surveillance is well-motivated, namely, to detect and prevent serious crimes such as terrorism and cyber-attacks on critical infrastructure. Protecting citizens from harm is indeed the first duty of government, and in a civilised society individuals have to some extent to be accountable to society as a whole. Privacy is therefore not an absolute right, but has to be balanced against other requirements, such as societal security. But the revelations raise major questions for society about how the masses of data about us recorded by computers every day can be used.

This impasse against which we find ourselves – that is, the tension between individual privacy and societal security – will quickly become much worse, as technology continues to permeate all aspects of human lives. Big data, the internet of things, and brain-computer interfaces mean that soon our every thought and action will be recorded by computer, with the consequent degradation of our hopes for privacy.

The challenge: A major challenge for society is to find ways to reconcile the requirement of societal security and that of individual privacy; and when that's impossible, to balance them appropriately. Society needs to agree a set of principles that govern when and how data about communications, finance, and internet usage should be used for preventing and detecting crime; these principles must express the sort of balance that society wants to have between the conflicting requirements. Rogaway's statement [9] distinguishes mass surveillance, which he condemns, and targeted surveillance, which he accepts. Other principles include the idea that surveillance should be proportionate, and *prima facie* justified.

It is appealing to rely on authorities to consider requests for data, one by one, taking into consideration the *prima facie* case for targeted surveillance. This approach requires unconditional trust in the parties that decide whether individual requests adhere to the agreed principles and rules, and citizens have no means to verify statements those parties may make about the extent and nature of the granted requests.

The vision: We propose a framework in which citizens can obtain verifiable and quantitative measures about the scale and nature of the surveillance that actually takes place. Under the proposal, there would still be legislation and procedures for determining whether access is allowed, on a case-by-case basis; but it would be supported by quantitative information about actual access that take place, against which citizens can hold politicians accountable. Based on this information, citizens can vote for governments and officers that demonstrate proportionality in the way they use

the data.

"Verifiable evidence" means that citizens have a means to check the veracity of the levels of surveillance that are reported. This is achieved using cryptographic protocols that produce data which can be subjected to tests by citizens. In principle, any citizen can verify the data, although it might be technically difficult and/or expensive to do so. It is sufficient if some trustworthy organisations (such as universities, charities, or journalists) do so on behalf of everyone else. This idea that users have software they can use to verify how data is handled remotely has been used elsewhere, e.g. in electronic voting. Incorrect actions by an election manager can be detected by tests that voters use to detect election integrity [39], [29] or voter coercion [34].

We aim in this paper to demonstrate one way in which this vision can be achieved, using what we call "accountable escrow". This means that governments are allowed to decrypt information about individuals, but there are mechanisms which make them accountable to citizens about the nature and quantity of the decryptions. Our system does not support *key escrow*: it does not allow authorities to recover a user's key. Rather, it supports *capability escrow*, in which escrow agents can use their own keys to perform decryptions. We say more about this distinction in section 1.1.

We study email communication, or more generally, person-to-person messaging. Our focus is on protecting the email content. Protecting the metadata is also important, but known to be much harder, and it is not the target here. Email content protection is already very controversial. The secure email services including Lavabit, PrivateSky, and Silent Circle have been shut down [12], [7], [15], or even been forced to stay up in order to continue collecting data through backdoors [14].

According to the Article 8 of the European Convention on Human Rights, privacy should be respected except as is in accordance with the law and is necessary in a democratic society in the interests of national security, public safety, etc. Every developed democracy in the world has laws that mandate Internet Service Providers have facilities to work with law enforcement for the purposes of lawful interception. In the UK law, Section 12 of RIPA imposes on Communication Service Providers obligations to ensure they are and remain able to provide an interception capability in order to give effect to interception warrants. The secure email service PrivateSky intended to provide truely end-to-end encryption [16], but this violated UK law and it had to shut down [19].

Society is best served by cooperating with democratically elected governments, and motivating them to protect our privacy while retaining the ability for crime prevention. Therefore it seems worthwhile to explore whether an email service that works along the lines of our "vision" above is possible. Such a service allows some surveillance but makes the quantity of surveillance evident to users.

Contribution: We design a balanced crypto system for asynchronous communication. Our contribution includes:

- We propose a general idea of *accountable escrow* and motivate this idea as an approach to reconciling and balancing the requirements of individual privacy and societal security.
- We present a novel method for escrowing decryption capability for public-key cryptography.
- We discuss and prove the computational security of our escrowed encryption under several variants of the Diffie-Hellman assumption.
- We formalise and prove the non-interactive escrow property in the Dolev-Yao model.
- We detail protocols for re-randomising government's decryption requests to protect users' privacy and confidentiality of government investigations.

1.1 Related work

Clipper Chip: The Clipper chip was a chipset that was developed and promoted by the NSA as an encryption device, in which a master key was held "in escrow" for release to law enforcement agencies. If government agencies established their authority to read or listen to a communication, then the key would be given to those government agencies, who could then decrypt all data transmitted by that particular device.

In contrast with our work, the Clipper chip had no accountability in how the escrowed key was used. Users could have no idea about the extent to which decryption of their communications took place. Moreover, clipper chip's key escrow capability was shown to be broken [25]. Another marked difference with our work is that the Clipper chip was intended to be mandatory. For this reason, the Clipper chip was not embraced by consumers or manufacturers and by 1996 the project had ceased to be active.

Key Escrow: Research in "key escrow" was popular in the 1990s; for example, [37], [47], [43]. In key escrow, the goal is to provide mechanisms allowing covert access to users' decryption keys. Key escrow was intended to be mandatory and ubiquitous across all kinds of encryption use-cases, and included recovery of session keys and signature keys. However, as criticised by Bruce Schneier and ten other researchers in [23], key recovery systems are inherently less secure, more costly, and more difficult to use than similar systems without a recovery feature.

In comparison, we provide an *optional* encryption method for email, which does not compromise or weaken any cryptography. Our proposal involves *accountable capability escrow* rather than covert key recovery. Our system does not recover any user's private key, and the private key is chosen by its owner and never leaves the owner. The

escrow agents answer requests without making any judgment about their appropriateness or validity (indeed, they have no knowledge about which users are referred to in the requests, or the corresponding ciphertext or plaintext). But they log the fact that they answer the request, in order to achieve accountability: citizens can monitor the quantity of government demands in real time.

There was also some work on identity escrow [38] which was about escrowing users' identity, instead of keys. The main discussion there was about how to issue and check the certificate of user's identifier in an escrowed manner and how to recover user's identity from escrowed certificate.

Identity-Based Encryption (IBE): In the IBE scheme [26], all the users' private keys are generated from a master key. Hence IBE has inherent key escrow, although it is not designed for that. In the IBE scheme, each user's private key is bounded with a particular identity string (e.g., email address). In comparison, our system let users choose their private keys freely, and thus the secure channels for key distribution needed by IBE are not needed. Our system does not require that users authenticate themselves with the government, or use government-issued identities, thus users can maintain pseudonymity. While it is very hard for IBE to fix key revocation problems, our system allows users choose a new secret key freely when their old key is compromised. It is usually considered that the use of IBE may be restricted to small, closed group or to applications with limited security requirements.

Time-releasing crypto system: Time-releasing cryptography introduces time delay to key escrow systems. For example, [24] splits the user's key into two parts: one part is escrowed with escrow agent and the other part is short and can be brute-forced. In this way, it delays the key recovery by requiring government to put some computational effort to find part of the secret key.

It is interesting to explore whether one can use time delay to reduce the trust of the trustees or other participants. Although useful in other contexts, this idea is not suitable for restricting law enforcement. Firstly, slowing down the decryption doesn't achieve the goal of accountability, and doesn't even give a meaningful restriction on the quantity of decryption allowed. Available computational power is increasing continuously, and trying to control the quantity of data that can be decrypted and examined through computational limits is likely to be fragile. It may cause a race between the search space for brute forcing being made larger, and law enforcement acquiring more computational power. Secondly, governments would find it hard to deploy a time-delay-based system. Law enforcement is already concerned about the delay of procedure for obtaining a court order. Moreover, in urgent crime investigations (like kidnapping), time delays may severely and artificially hamper investigations.

In comparison, our accountable escrow system enables escrow agency to decrypt for government blindly and timely while making the quantity of government's demands evident.

1.2 Outline

Section 2 presents the design of an accountable escrow crypto system for asynchronous communication. This section includes: Section 2.1 describes agents in our system; Section 2.2 discusses the difficulty of design; Section 2.3 presents our escrow public-key cryptography; Section 2.4 details protocols for government decryption requests; Section 2.5 describes the public audit log.

Section 3 discuss and proves the computational security of our escrowed encryption scheme. Section 4 formalises and proves the non-interactive escrow property of our system in Dolev-Yao model. Section 5 gives a general discussion about the granularity of accountability. Section 6 envisions the other application areas of the concept of balancing security and privacy. The paper concludes in Section 7.

2 Accountable escrow for asynchronous communication

If Bob wants to send a message to Alice, he currently has the following three options:

1) He can send the message insecurely in plaintext (this is currently the most popular option);
2) He can use S/MIME or OpenPGP to manage his friends' public-keys on his own to obtain end-to-end encryption. This gives him maximal security, but is very unpopular due to the bad usability [46].
3) He can use a packaged up encrypted email service such as Lavabit, Silent Circle, PrivateSky and Dark Mail, if any of these services can be trusted and is still available [12], [14], [7], [15].

The system we propose is not intended to replace any of these options. Rather, it is intended to offer as a fourth alternative which aims to motivate government to directly support, or at least not to thwart commercial attempts at, building a convenient and usable public key infrastructure for secure messaging.

Our system provides *accountably-escrowed encryption* for public-key cryptography. Here, *escrowed encryption* means that the system offers a sender a securely certified public key which allows decryption by the receiver, and also decryption-with-accountability by the government.

Users may be motivated to use our system because it gives them a convenient public-key infrastructure, the use of which prevents snooping from service providers, foreign governments, and other third party attackers, while having domestic government support because of the accountable

escrow. Companies may be motivated to build the system because of domestic government support and user demand. This includes issuing and transparently certifying [20], [41] public keys for individuals and producing usable tamper-resistant devices for storing keys.

The design of our system uses public-key systems based on *discrete logarithm problem*. The idea can also be deployed to the elliptic curve cryptography.

The public parameters of our system are given by $(\mathbb{G}, q, g)$ where $\mathbb{G}$ is a cyclic group with a generator g, and of prime order q. We assume our group does not support bilinear pairing which will be explained later. We use multiplicative notation '$\cdot$' for group operation. We define $\mathbb{Z}_q^* := \{1, \ldots, q-1\}$. We write $x \xleftarrow{R} V$ for randomly choosing x from V. We write g^x for the group element of $\mathbb{G}$ that results from multiplying x copies of g. We can see that $\prod_{i \in I} g^{x_i} = g^{\sum_{i \in I} x_i}$.

2.1 Agents

There are four parties involved in our system:

Users are the common people who want to communicate with each other. Each user has a private key and an *escrow public key*.

Government (Gov) wants to decrypt the users' messages for the purpose of investigation and surveillance.

Custodians (Cust) are a predefined list of distributed trusted third-party escrow authorities: $\mathsf{Cust}_1, \mathsf{Cust}_2, \cdots, \mathsf{Cust}_n$. Each custodian Cust_i is equipped with a unique private key/public key pair (c_i, g^{c_i}). Custodians perform decryptions for government requests and put each decryption request into an audit log.

Each user chooses a set of custodians that he is willing to trust and escrows decryption capability of his private key with the custodians. The custodians jointly perform the decryption and none of the custodians can decrypt on its own.

Certificate Authority (CA) checks the correctness of escrow in user's registration, issues and certifies the user's escrow public key. CA doesn't have to know any private information (e.g., name, email address, phone number) about an owner of a public key. CA can prove to government that it generates the escrow public key correctly. The trust of CA can also be reduced by certificate transparency [20], [41]. The CA here is for individual users, and the custodians' public keys are certified by the traditional CA.

2.2 Problems

The main difficulty of designing such a balanced system arises from the conflicts of interest between the involved parties. For example, users want to communicate privately with each other, so they do not want any irrelevant people to see their conversation, while governments want to analyse everyone's messages without anyone knowing. Because of the conflicts of interest, the difficulty of designing such a system is to enforce that each party complies with protocol, rather than expecting them to cooperate. As a matter of principle, we do not want to introduce any censorship or backdoor.

In order to end-to-end encrypt asynchronous communication (e.g., email), every user has their own private key/public key pair. The problem is that how we can enable custodians to decrypt the messages which are encrypted with each user's own public key.

To encrypt a message for multiple recipients, we can adapt a 3-party Diffie-Hellman protocol into a 2-party ElGamal encryption scheme. Assume two recipients Alice and her custodians each own a private key a and c, and have a joint public key $PK_{AC} = (g^a,\ g^c,\ g^{ac})$. Now a sender Bob is supposed to encrypt a message m for both Alice and her custodians by producing the encryption $(g^{ar},\ g^{cr},\ m \cdot g^{acr})$. After receiving the message, Alice can first compute g^{acr} by applying her private key a to g^{cr} and then obtain m by computing $m \cdot g^{acr} / g^{acr}$. Similarly custodians can obtain m by applying their key c to g^{ar}. In this way, the communication between Alice and Bob is encrypted – both their service providers and the other third-party attackers cannot snoop their conversation; meanwhile, if government wants to investigate this message, they can simply get the ciphertext $(g^{ar},\ g^{cr},\ m \cdot g^{acr})$ from service providers (or from network tapping) and send g^{ar} to custodians for decryption.

However, this kind of escrow can be easily bypassed. The joint public key PK_{AC} can be used to derive two "pure" public keys for Alice: g^a and $(g^c,\ g^{ac})$. Now assume Bob dislikes the custodians and he only wants Alice to get the message. Then Bob can encrypt m as $M = (g^r,\ m \cdot g^{ar})$ or $(g^{cr},\ m \cdot g^{acr})$ which prevents custodians from decrypting and obtaining message m. Even if we introduce some auditing mechanism (e.g., zero-knowledge proof) to ensure the encrypted packets are correctly generated, users can still bypass the escrow by double encryption.

While our system decides what type of public key it is going to certify and publish in its public key infrastructure, the choice of client software for cryptographic operations is up to the users. Users may use any software they find on the web. The above weakness can be easily implemented as a plugin to the client software by some hackers. Users can simply download and install such a plugin to change the underlying encryption and decryption algorithms to circumvent each other's custodians. We shall further discuss this point in the following Section 4.

The escrow in Escrow ElGamal [26] and Tripartite Diffie-Hellman protocol [36] can be easily bypassed (although

these two papers do not have escrow as a target), because it reveals each user's pure public key xP where x is the user's private key. The escrow in both schemes is constructed by using bilinear pairing. Even if we consider $\hat{\mathsf{e}}(xP, sP)$ as joint public key for the user, where (s, sP) is the escrow agency's private key/public key pair, it can still be bypassed by using $(\hat{\mathsf{e}}(P, sP), \hat{\mathsf{e}}(P, sP)^x)$ as the user's public key. This is because P and sP are the publicly known information, and $\hat{\mathsf{e}}(xP, sP) = \hat{\mathsf{e}}(P, sP)^x$. The encryption under this public key can only be decrypted by the user who knows x.

2.3 Escrow public-key system

We propose a novel method for escrow in public-key cryptography. Each user escrows the decryption capability of his private key to custodians through a special public key, called *escrow public key*. The private key itself is not escrowed. All the decryption requests have to go through custodians which holds government accountable. The custodians can decrypt by simply applying their own private keys to the ciphertext, and it does not involve using any user's information, e.g., the user's private key or public key. Hence our escrow scheme is very easy to manage and at no extra costs.

Escrow Public Key Registration: The registration phase is shown in Figure 1. The user's *escrow public key* is generated as below:

1) Alice chooses her private key a, and several custodians $\{\mathsf{Cust}_i\}_{i\in I}$ she is willing to trust. Assume each Cust_i has a private key/public key pair (c_i, g^{c_i}). Let $c := \sum_{i\in I} c_i$ be these custodians' shared private key. Then g^c is their shared public key computed by $\prod_{i\in I} g^{c_i}$. The distributed key scheme we use for custodians is the simplest one from [27], although it could also be a threshold scheme [44].
2) Alice computes (g^a, g^c, g^{ac}) and sends them to CA. The correctness of applying a to some valid custodians' shared public key g^c can be proved by a ZKP [30]. In order to make sure CA can prove to government the correctness of escrow, we can use a non-interactive ZKP in this step.
3) If the proof is correct, CA chooses a random s and computes Alice's *escrow public key*:

$$ePK_A := \left(g^s,\ g^{s(a+c)},\ g^{sac}\right)$$

certifies and publishes it. The correctness of applying s to $(g,\ g^{a+c},\ g^{ac})$ can be proved by a ZKP [30].

Accountably-Escrowed Encryption (AEE): When Bob wants to send Alice a message m, he chooses a random r and encrypts m with Alice's escrow public key ePK_A as a ciphertext $C := (C_1, C_2, C_3)$ where

$$C_1 := g^{sr},\ C_2 := g^{s(a+c)r},\ C_3 := m \cdot g^{sacr}$$

After Alice receives the message, she computes m by

$$m = \frac{C_3}{\left(\frac{C_2}{{C_1}^a}\right)^a}$$

The decryption with the custodians' shared private key c is similar, but uses c instead of a. We shall detail several decryption protocols in the following Section 2.4.

We will discuss and prove the computational security of our AEE scheme in the following Section 3.

Notice that each part of Alice's public key ePK_A is completely symmetric to Alice and her custodians. Intuitively, this means whatever the computations Alice can perform on the encryption sent from Bob, her custodians can also compute similarly. We shall formalise and prove this property in Dolev-Yao model in the following Section 4.

As illustrated in Section 2.2, the bilinear-pairing-based escrow can be easily bypassed. In contrast, our capability escrow is not constructed by bilinear pairing. Furthermore, the group used in our system does not support bilinear pairing; otherwise users can bypass the escrow by constructing a pure public key $(\hat{\mathsf{e}}(g^s, g^c), \hat{\mathsf{e}}(g^s, g^c)^a)$ for Alice, given g, g^c are publicly known information and g^s, g^{sac} are part of Alice's escrow public key and $\hat{\mathsf{e}}(g^s, g^c)^a = \hat{\mathsf{e}}(g, g^{sac})$.

2.4 Protocols for blind decryption by custodians

Since the decryption capability of the user's private key is escrowed to custodians, one major concern is that the government's decryption requests might leak information about the identities of suspects and the details of messages that they look into. Although the custodians are supposed to be trustworthy, we want to minimise the trust as much as possible to protect the user's privacy and confidentiality of government investigations.

First of all, to decrypt a message, it is not necessary to send to custodians the entire message packet including the meta data (e.g., header of email) and the whole encrypted message body. In fact, it is sufficient to only submit some parts of the encryption, e.g., C_1, C_2 described in Section 2.3. In the following discussion, we first describe a naive decryption protocol which leaks some details about government's investigation. Then we detail two re-randomisation methods to improve the protocol to fully protect the user's privacy and confidentiality of investigations.

To simplify the description, we assume that government communicates with custodians through some secure channels. The secure channels are easy between government and custodians because they all have well-established traditional public keys. In several cases, we can get rid of the secure channels. One case is when government is also a custodian, i.e. $c = k + c'$ where k is government's private key and c' is the shared private key of the other custodians, then anyone who wants to decrypt the message has to obtain

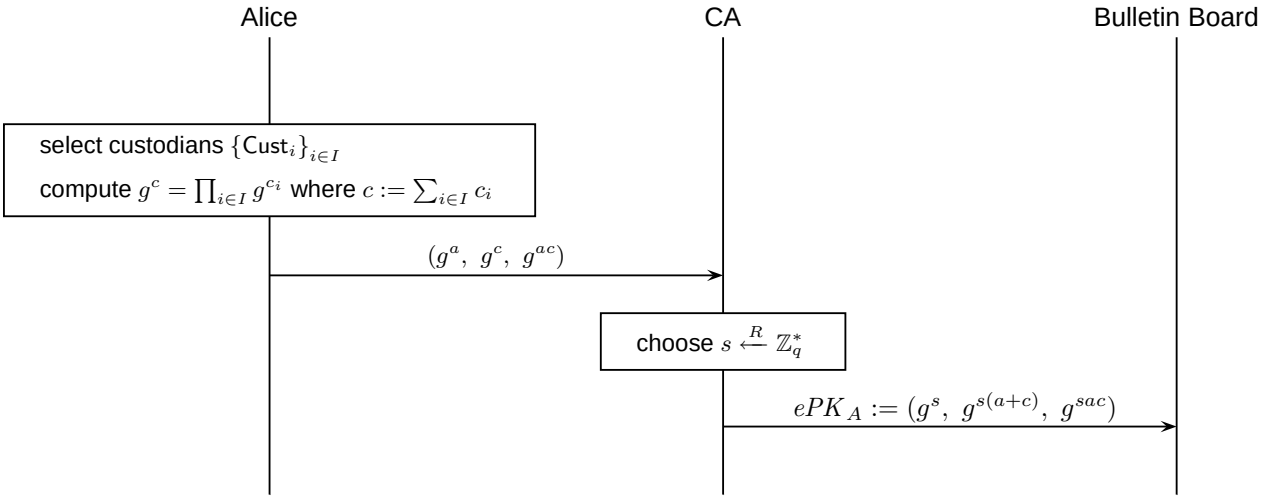

Figure 1: Escrow Public Key Registration

government's secret share which is impossible to get by eavesdropping the conversation between government and custodians. Another case that the secure channels are not necessary is when we re-randomise the decryption requests which will be discussed later.

A naive protocol of handling government's decryption requests is presented in Figure 2. The decryption in this protocol has two phases:

1) In the first phase, government sends g^{sr} to each of Alice's custodians Cust_i. Each Cust_i verifies government's signature and applies its private key c_i and generates a ZKP to prove that c_i has been correctly applied.
2) After receiving all the secret shares for decryption, government computes g^{sar}. The second phase is exactly the same as the first phase except sending g^{sar} instead of g^{sr}. At the end of the second phase, government gets g^{sacr} and then decrypts the message.

In this naive protocol, if the custodians log the term g^{sr} or the pair (g^{sr}, g^{sar}) in the public log, then the sender and the receiver of the message and any third party attacker who has eavesdropped Bob and Alice's conversation before would know that they are under investigation. The information leakage of investigations could alert criminals to the fact of an investigation concerning them, or being known as a suspect could cause the other people's prejudice. One possible solution is to let custodians modify the data before putting them into a public audit log, but this relies on the trust of all the custodians to keep the original data secret and carefully disguise the data.

In fact, there are better ways to guarantee the decryption is successful but the details of the data are not identifiable. Our system allows government to decrypt a message without leaking any detailed information to the custodians, the sender or the receiver of the message, or the other third party attackers. The custodians are able to perform *blind decryption* for government, thus the user's privacy and the confidentiality of investigations are fully protected while it does not rely on the trust of any custodian. Meanwhile, custodians put each decryption request from government into a public log and users can monitor this log to get the total number of decryption requests in real time.

To decrypt $\left(g^{sr},\ g^{s(a+c)r},\ m \cdot g^{sacr}\right)$, there are two ways to re-randomise the decryption request. The first method is described in Figure 3. In this method, government chooses a random x and computes g^{xsr}. After receiving g^{srxc_i}, government applies x^{-1} to eliminate x and gets g^{src_i}.

The second method for re-randomisation is given in Figure 4 which is similar to the re-encryption of ElGamal encryption [32]. In this method, government also chooses a random y, computes g^{cy} and multiplies g^{sr} to g^y. After collecting all the $g^{(sr+y)c_i}$, the factor y can be eliminated by dividing g^{cy}.

The Figure 3 and Figure 4 give the details for re-randomisation of decryption of phase I, and the re-randomisation for the second phase of the decryption is similar. We can also mix these two methods in Figure 3 and Figure 4 across the two phases.

Moreover, re-randomisation can also help us get rid of secure communication channels between government and

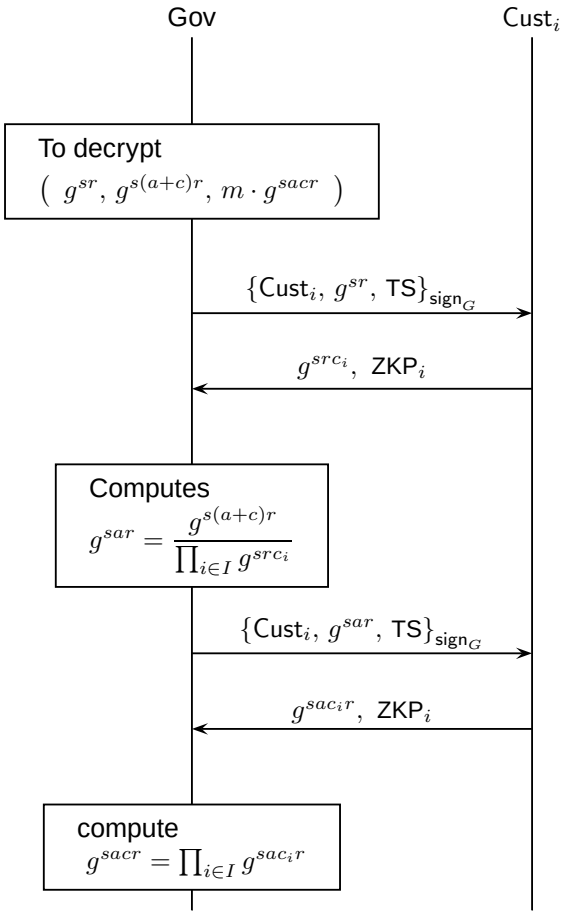

Figure 2: Naive Decryption Protocol. TS denotes timestamp, and sign_G denotes government's signature. ZKP_i is used to prove that Cust_i correctly applies their private key c_i.

custodians because the decryption requests are actually "encrypted" by the randomisation factor x.

2.5 Accountability: the public audit log

The custodians jointly or separately maintain an audit log for all the decryption requests they received from government. Because each decryption requires all the secret shares of the involved custodians, we only need to trust that there exists at least one custodian who will be honest and will document the request into the log. Because the decryption requests are re-randomised, custodians can publish these decryption requests in real time without leaking any detailed information about investigations.

The log is organised as an append-only Merkle tree [42]. A Merkle tree is a binary tree whose nodes are labelled by hash values. Leaf nodes are labelled by the hashes of decryption requests that have been appended to the log. Every non-leaf node is labelled with the hash of the labels of its two children nodes. Merkle hash trees can prove two things very efficiently: one is that appended decryption requests have not later been removed from the log (this is also called the append-only property); and the other one is that a particular decryption request has been appended to the log.

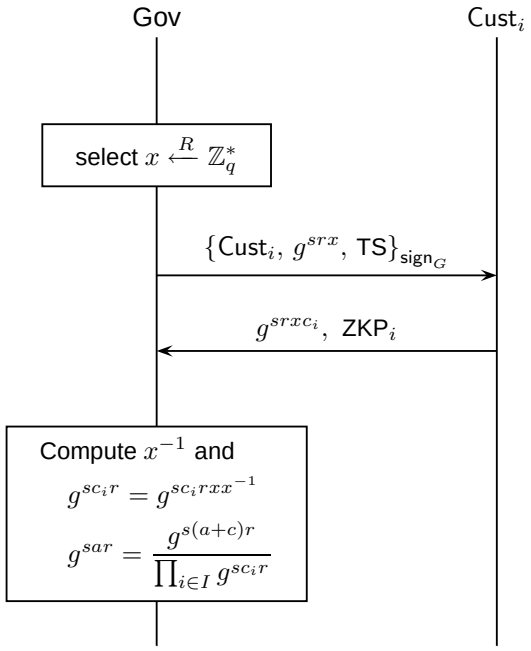

Figure 3: Re-randomised Decryption – Method I

Whenever a custodian Cust receives a decryption request of the form $\{\mathsf{Cust}, X, \mathsf{TS}\}_{\mathsf{sign}_G}$, he checks signature and timestamp TS (avoiding replay attack). If the checks hold, he puts this request into the append-only public log for auditing. Hence monitors (who keep a entire lists of decryption requests that are in a log) and users can regularly verify that logs are behaving properly.

The structure of the log is thus similar to the log of public certificates used in certificate transparency [20], except that the data items are the re-randomised decryption requests instead of public key certificates. The Merkle tree properties of the log make it impossible for a malicious log maintainer to remove data or otherwise tamper with the log.

The decryption of each message involves two decryption requests $\{\mathsf{Cust}, X_1, \mathsf{TS}_1\}_{\mathsf{sign}_G}$ and $\{\mathsf{Cust}, X_2, \mathsf{TS}_2\}_{\mathsf{sign}_G}$ where $X_1 = g^{srx}$ or g^{sr+x}, and $X_2 = g^{sarz}$ or g^{sar+z}. Note that x and z are different random elements; otherwise the

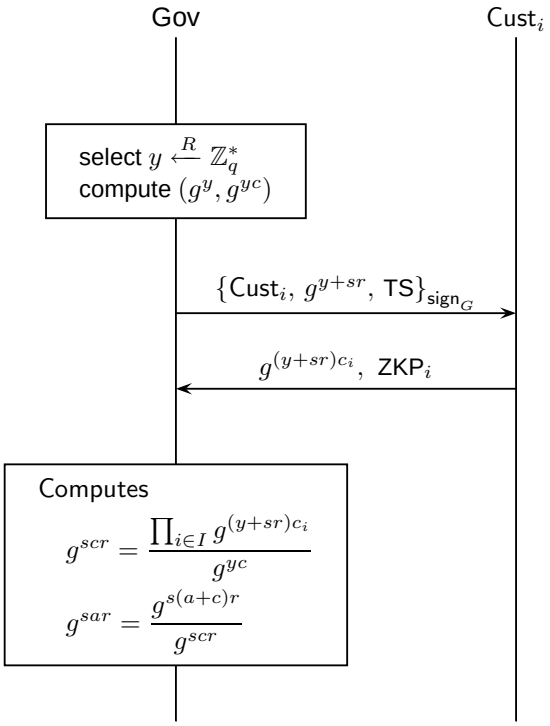

Figure 4: Re-randomised Decryption – Method II

log would allow user-traceability. For example, if $X_1 = g^{srx}$ and $X_2 = g^{srax}$, then Alice can know whether this is her message by testing $X_1^a =^? X_2$; if $X_1 = g^{sr+x}$ and $X_2 = g^{sra+x}$, Alice can first compute $Y \leftarrow (X_1^a/X_2)^{((a-1)^{-1})}$ and then test $(X_1/Y)^a =^? X_2/Y$. The monitors and users can count the total number of decryption requests as $N/2$ where N is the total number of different pairs (X, TS).

While the confidentiality of the decryption requests is protected, citizens can see how many government demands occur in any particular period, such as an hour, a day, or a year. Using this mechanism, government's investigation and surveillance is made accountable to citizens. People can decide through the democratic process how much of investigation and surveillance they want to allow, and the quantity is verifiable by citizens. This meets current appeal for government surveillance reform [17] which asks for the quantity and nature of government demands to be published.

A feature of our system is that the custodians cannot provide the government with a key that would allow covert decryption of a particular user's data. All they could do is hand over their entire secret key, making themselves completely redundant. This is in marked contrast with key escrow systems [37], [47], [43], where the job of escrow agents is to provide users' keys to the government for covert decryption.

3 Computational security of AEE scheme

In this section, we discuss the computational security of our AEE scheme proposed in Section 2.3. Recall that our AEE scheme makes use of a cyclic group $\mathbb{G}$ of prime order q with a generator g. The escrow public key for a user A is

$$ePK_A := \left(g^s,\ g^{s(a+c)},\ g^{sac}\right)$$

To encrypt a message m, one chooses a random $r \xleftarrow{R} \mathbb{Z}_q^*$, computes:

$$\left(g^{sr}, g^{s(a+c)r}, m \cdot g^{sacr}\right)$$

The *Chosen-Ciphertext Security* of the scheme AEE is defined by the following chosen-ciphertext attack game, played between an adversary $\mathcal{A}$ and a challenger $\mathcal{B}$:

1) **Setup**. $\mathcal{B}$ sets up the secret keys $sk = (x, y)$ and the escrow public key *ePK* and sends *ePK* to $\mathcal{A}$.
2) **Phase 1**. $\mathcal{A}$ makes a number of decryption queries to $\mathcal{B}$, where the input to each query is a ciphertext, say C. To answer such a query, $\mathcal{B}$ decrypts C by using either x or y and sends the result to $\mathcal{A}$.
3) **Challenge**. Once $\mathcal{A}$ decides that Phase 1 is over, it outputs two equal length plaintexts $m_0, m_1 \in \mathcal{M}$ on which it wishes to be challenged. $\mathcal{B}$ picks a random bit $b \in \{0, 1\}$, encrypts m_b, and sends the resulting ciphertext C^* as the challenge to $\mathcal{A}$.
4) **Phase 2**. $\mathcal{A}$ issues more decryption queries as in Phase 1, but with the restriction that $C \neq C^*$.
5) **Guess**. Finally, $\mathcal{A}$ outputs a guess $b' \in \{0, 1\}$ and wins the game if $b = b'$.

We refer to $\mathcal{A}$ as an IND-CCA adversary and define its advantage over the scheme AEE by

$$\mathsf{AdvCCA}_{\mathcal{A},\mathsf{AEE}} = \left|\Pr[b = b'] - \frac{1}{2}\right|$$

The probability is over the random bits used by $\mathcal{A}$ and $\mathcal{B}$.

Definition 1. *We say that an AEE scheme is* IND-CCA *secure if for any probabilistic polynomial time* IND-CCA *adversary $\mathcal{A}$ the advantage* $\mathsf{AdvCCA}_{\mathcal{A},\mathsf{AEE}}$ *is negligible.*

If the adversary is not allowed to ask any decryption query, then the above game defines security of an AEE scheme again Chosen-Plaintext Attacks (CPA). We refer to $\mathcal{A}$ as an IND-CPA adversary and define its advantage over the scheme AEE by

$$\mathsf{AdvCPA}_{\mathcal{A},\mathsf{AEE}}(\kappa) = \left|\Pr[b = b'] - \frac{1}{2}\right|.$$

Definition 2. *We say that an AEE scheme is* IND-CPA *secure if for any probabilistic polynomial time* IND-CPA *adversary $\mathcal{A}$ the advantage* $\mathsf{AdvCPA}_{\mathcal{A},\mathsf{AEE}}(\kappa)$ *is negligible.*

The *Decisional Diffie-Hellman assumption (DDH)* states that two randomly generated (g^x, g^y, g^{xy}) and (g^x, g^y, g^z) are computationally indistinguishable. We extend this assumption from 2-party to 3-party and call it *3-Party Decisional Diffie-Hellman assumption (3-DDH)*:

it is hard to distinguish $(g, g^x, g^y, g^{xy}, g^z, g^{z(x+y)}, g^{zxy})$ from $(g, g^x, g^y, g^{xy}, g^z, g^{z(x+y)}, g^u)$, where u is randomly chosen.

Similar multi-party Diffie-Hellman problems are also discussed in [45], [28]

The AEE encryption scheme is malleable and it has similar security with ElGamal encryption [32]. We start by proving the scheme AEE is secure under chosen-plaintext attack and then discuss several existing techniques to upgrade it into a secure scheme under chosen-ciphertext attack.

Theorem 3. *The AEE encryption scheme is IND-CPA secure under the 3-DDH assumption.*

Proof Sketch: Let $\mathcal{A}$ be an adversary who breaks the AEE scheme in the sense of IND-CPA. We consider a distinguisher D who breaks the 3-DDH assumption using $\mathcal{A}$ as follows.

D takes randomly generated $U_0 = (g, g^x, g^y, g^{xy}, g^z, g^{z(x+y)}, g^{zxy})$ or $U_1 = (g, g^x, g^y, g^{xy}, g^z, g^{z(x+y)}, g^u)$ as input. We denote this input by $(g, g^x, g^y, g^{xy}, g^z, g^{z(x+y)}, K)$, where $K = g^{zxy}$ or g^u.

Setup: To start the game, D chooses $w \xleftarrow{R} \mathbb{Z}_q^*$ and constructs an escrow public key as $ePK = (g^w, g^{w(x+y)}, g^{wxy})$. D gives ePK to $\mathcal{A}$.

Challenge: $\mathcal{A}$ outputs two messages m_0, m_1 on which it wishes to be challenged. D chooses a random bit $b = \{0, 1\}$ and sets $C_1^* := g^z$, $C_2^* := g^{z(x+y)}$, $C_3^* = m_b \cdot K$ and returns the ciphertext $C^* = (C_1^*, C_2^*, C_3^*)$ to $\mathcal{A}$. Note that z can be regarded as rw for some hidden r.

Guess: $\mathcal{A}$ outputs its guess b' for b. D outputs b' as well.

If $K = g^{zxy}$, then C^* is an encryption of m_b and $\mathsf{AdvCPA}_{\mathcal{A},\mathsf{AEE}}(\kappa) = \left|\Pr(D(U_0) = 1) - \frac{1}{2}\right|$; otherwise if $K = g^u$ was chosen randomly, C^* contains no information about the message from the adversary's view point, and $\Pr(D(U_1) = 1) = \frac{1}{2}$. Therefore, $\mathsf{AdvCPA}_{\mathcal{A},\mathsf{AEE}}(\kappa) = |\Pr(D(U_0) = 1) - \Pr(D(U_1) = 1)|$ and the theorem follows.

■

There are many existing techniques for upgrading a CPA-secure scheme into a CCA-secure scheme, for example the DHIES scheme [22] and the Fujisaki-Okamoto generic transformation [33].

We consider a modified DHIES scheme presented in [40] in which the MAC is eliminated from DHIES. Let $SE = (\mathcal{E}, \mathcal{D})$ be an IND-CCA secure symmetric encryption scheme and H is a secure hash function. We call the following encryption scheme *hash-AEE*:

$$C_1 := g^{sr},\ C_2 := g^{sr(a+c)},\ K := H(C_1, C_2, g^{rsac})$$
$$C_3 := \mathcal{E}(K, m)$$

The ciphertext is (C_1, C_2, C_3).

The Oracle Diffie-Hellman (ODH) assumption, as described in [22], [40], claims that two randomly generated $(g, g^x, g^y, H(g^{xy}))$ and (g, g^x, g^y, r), where the length of hash output is $|H|$ and $r \xleftarrow{R} \{0,1\}^{|H|}$, are computationally indistinguishable even if a distinguisher D is given access to an oracle $\mathcal{H}_x$ where $\mathcal{H}_x$ returns $H(u^x)$ for a query u. In this definition, D is not allowed to query g^y to $\mathcal{H}_x$.

We extend the ODH assumption from 2-party ODH to 3-party and call it *3-party Oracle Diffie-Hellman (3-ODH)*: Two randomly generated $(g, g^x, g^y, g^{xy}, g^z, g^{(x+y)z}, H(g^z, g^{(x+y)z}, g^{xyz}))$ and $(g, g^x, g^y, g^{xy}, g^z, g^{(x+y)z}, r)$, where the length of hash output is $|H|$ and $r \xleftarrow{R} \{0,1\}^{|H|}$, are computationally indistinguishable even if a distinguisher D is given access to two oracles $\mathcal{H}_\alpha$ where α is either x or y and $\mathcal{H}_\alpha$ returns $H(u, v, (v/u^\alpha)^\alpha)$ for a query (u, v). In this definition, D is not allowed to query its input $(g^z, g^{(x+y)z})$ to $\mathcal{H}_x$ or $\mathcal{H}_y$.

Theorem 4. *The hash-AEE encryption is IND-CCA secure under the 3-ODH assumption if the underlying symmetric cipher is IND-CCA secure.*

Due to space limitation, the proof can be found in the full version of this paper.

4 The escrow property in Dolev-Yao model

Our system provides each user an escrow public key in its public key infrastructure, but the choice of client software for cryptographic operations is up to the users. As mentioned before, a user may use any software he found on the web. That is to say, the encryption and decryption algorithms are not fixed. Hence, the problem is whether the users can bypass the escrow by changing the underlying software, as described in Section 2.2. We formalise this problem in Dolev-Yao model and prove that our escrow is enforced, that is, *by only using Alice's escrow public key and publicly known information, Bob cannot compute any encrypted packet which Alice can decrypt but not her custodians.*

Intuitively, every part of our escrow public key is completely symmetric to Alice and her custodians which indicates whatever the computations Alice can perform her

custodians can perform similarly. However, we need to prove the result formally because this intuition is not always valid. Consider the following "symmetric" public key:

$$PK_A := (g, g^{a+c}, g^{ac})$$

It is as symmetric as our ePK_A and the only difference is that PK_A is not re-randomised by a random s. However, the key escrow under PK_A can be easily bypassed. Given the Custodians' public key g^c as common knowledge, Bob can derive g^a from PK_A by computing g^{a+c}/g^c and then encrypt a message m as $(g^r, m \cdot g^{ra})$ which custodians are not capable of decrypting. This explains why we need a random s in ePK_A.

We shall formalise our problem in the Dolev-Yao attacker model which assumes that the attacker can intercept and forge messages using a given set of cryptographic primitives. Cryptographic operations are treated abstractly as black boxes of which the computational detailed features are abstracted away.

4.1 Basic definitions

We follow the classical notation and terminology for equational theory [21] and term rewriting [31]. A signature Σ consists of a finite set of function symbols each with an arity. A function symbol with arity 0 is a constant symbol. Given a signature Σ, an infinite set of names $\mathcal{N}$, and an infinite set of variables $\mathcal{X}$. *Terms* L, M, N are constructed by names, variables and function applications. A *term context*, ranged over by $\mathcal{C}, \mathcal{D}$, is a term with several holes. A substitution $\sigma := \{M_1 \mapsto N_1, \ldots, M_k \mapsto N_k\}$ replaces terms $M_1, \ldots, M_k$ with $N_1, \ldots, N_k$.

Each signature Σ is equipped with an equivalence relation $=_\Sigma$ on terms that is closed under substitutions of terms for variables and one-to-one renamings and applications of term contexts. We may also write $=_E$ for the smallest equivalence relation that is a closure of a set of equations E and applications of term contexts.

A *term rewriting system* (TRS) is a finite set of rewriting rules $\{M \to N\}$ on terms. We write R_Σ for the rewriting system defined on Σ. We write $L_1 \to_{R_\Sigma} L_2$ when there exists a context $\mathcal{C}$ and a substitution σ such that $L_1 = \mathcal{C}[M\sigma]$ and $L_2 = \mathcal{C}[N\sigma]$ and $M \to N \in R_\Sigma$. We define $\xrightarrow{*}_{R_\Sigma}$ to be the reflexive and transitive closure of $\to_{R_\Sigma}$. Given a set of rewrite rules R and a set of equations E, we define $\to_{R/E}$ as $=_E; \to_R; =_E$. A rewriting system $\mathcal{R}$ is *E-confluent*, iff for every M, N such that $M =_{\mathcal{R}\cup E} N$, there exists M' such that $M \xrightarrow{*}_{\mathcal{R}/E} M'$ and $N \xrightarrow{*}_{\mathcal{R}/E} M'$. $\mathcal{R}$ is E-convergent if, in addition, $\to_{R/E}$ is terminating.

4.2 Diffie-Hellman theory in Dolev-Yao model

We model Diffie-Hellman Theory by exponentiation $\mathsf{exp}(u, x)$, multiplication $x \cdot y$ and multiplicative inverse function $x^{-\mathbf{1}}$, an identity unit $\mathbf{1}$, and a group generator g:

$$DH := \left\{ \_ \cdot \_,\ \mathsf{exp}(\_,\_),\ \ \_^{-\mathbf{1}},\ \mathbf{1},\ g \right\}$$

Definition 5 (Equations E_{DH} for *DH*)**.** *The equations for Abelian Group are modelled by equations E_{AG}:*

$$\begin{gathered} x \cdot y = y \cdot x \\ x \cdot (y \cdot z) = (x \cdot y) \cdot z \\ x \cdot \mathbf{1} = x \qquad x^{-\mathbf{1}} \cdot x = \mathbf{1} \\ (x^{-\mathbf{1}})^{-\mathbf{1}} = x \qquad x^{-\mathbf{1}} \cdot y^{-\mathbf{1}} = (x \cdot y)^{-\mathbf{1}} \end{gathered}$$

We write u^x for $\mathsf{exp}(u, x)$ for short. The equations for Diffie-Hellman exponentiation are given by E_{Exp}:

$$\begin{gathered} \mathbf{1}^u = \mathbf{1} \quad u^{\mathbf{1}} = u \quad (u^{-\mathbf{1}})^x = (u^x)^{-\mathbf{1}} \\ (g^x)^y = g^{x \cdot y} \quad (x \cdot y)^z = x^z \cdot y^z \end{gathered}$$

We write E_{DH} for $E_{AG} \cup E_{Exp}$.

Although we have '+' in our public key, i.e. $g^{s(a+c)}$ and $g^c := g^{\sum_i c_i}$, it is essentially a multiplication of two exponentiations, i.e. $g^{s \cdot a} \cdot g^{s \cdot c}$ and $\prod_i g^{c_i}$.

Our proof technique can be generalised to a user-defined signature for modelling cryptographic operators. To simplify its presentation, we consider the classic cryptographic operations e.g., symmetric/asymmetric encryption, one-way hash function, pairs, signature, which are modelled as a set of function symbols *Crypt* with equations E_{Crypt}:

$$Crypt := \left\{ \begin{array}{c} \langle \_,\_ \rangle,\ \mathsf{fst}(\_),\ \mathsf{snd}(\_), \\ \mathsf{senc}(\_,\_,\_),\ \mathsf{sdec}(\_,\_),\ \mathsf{hash}(\_), \\ \mathsf{aenc}(\_,\_,\_),\ \mathsf{adec}(\_,\_),\ \mathsf{pk}(\_) \end{array} \right\}$$

$$E_{Crypt} := \left\{ \begin{array}{c} \mathsf{fst}(\langle x, y \rangle) = x \\ \mathsf{snd}(\langle x, y \rangle) = y \\ \mathsf{sdec}(\mathsf{senc}(x, y, z), y) = x \\ \mathsf{adec}(\mathsf{aenc}(x, \mathsf{pk}(y), z), y) = x \end{array} \right\}$$

These function symbols abstract the other cryptographic operations which are not based on Diffie-Hellman problems, for example AES, RSA-OAEP.

We define $=_{DHC}$ as the smallest reflexive, transitive and symmetric closure of $E_{DH} \cup E_{Crypt}$. and term contexts.

Definition 6 (Deducibility $(\mathcal{E}, T) \vdash M$)**.** *Given a set of terms T and a set of names $\mathcal{E}$, the deducibility of a given term M is defined by $(\mathcal{E}, T) \vdash M$:*

$$\frac{}{(\mathcal{E}, T) \vdash M} M \in T \qquad \frac{}{(\mathcal{E}, T) \vdash n} n \in \mathcal{N} \text{ and } n \notin \mathcal{E}$$

$$\frac{(\mathcal{E}, T) \vdash M_1, \ldots, (\mathcal{E}, T) \vdash M_k}{(\mathcal{E}, T) \vdash f(M_1, \ldots, M_k)}$$

$$\frac{(\mathcal{E}, T) \vdash M \quad M =_{DHC} N}{(\mathcal{E}, T) \vdash N}$$

Intuitively, $\mathcal{E}$ models the secrets like the secret key, and T models intruder's initial knowledge. $(\mathcal{E}, T) \vdash M$ models the

computations that the intruder can perform to deduce new messages. Because the equational theory $=_{DHC}$ is closed under context applications, we can see that

Proposition 7. *$(\mathcal{E}, T) \vdash M$ iff there exists a context $\mathcal{C}$ with $name(\mathcal{C}) \cap \mathcal{E} = \emptyset$ and a set of terms $M_1, \ldots, M_k \in T$ such that $\mathcal{C}[M_1, \ldots, M_k] =_{DHC} M$.*

Theorem 8. *Let $T = \{g,\ g^c,\ g^s,\ g^{sa} \cdot g^{sc},\ g^{sac}\}$ and $\mathcal{E} = \{s, a, c\}$. Let $\mathcal{E}_{Bob}$ be a set of names with $\mathcal{E}_{Bob} \cap \mathcal{E} = \emptyset$. Let $T_{Alice} = T \cup \{a\}$ and $T_{Cust} = T \cup \{c\}$.*

Let a term $M_{Bob} := \mathcal{C}[M_1, \ldots, M_k]$ where $\mathcal{C}$ is a term context with $name(\mathcal{C}) \cap (\mathcal{E} \cup \mathcal{E}_{Bob}) = \emptyset$ and terms $M_1, \ldots, M_k \in T \cup \mathcal{E}_{Bob}$. If $(\mathcal{E} \cup \mathcal{E}_{Bob}, T_{Alice} \cup \{M_{Bob}\}) \vdash m$ with $m \in \mathcal{E}_{Bob}$, then $(\mathcal{E} \cup \mathcal{E}_{Bob}, T_{Cust} \cup \{M_{Bob}\}) \vdash m$.

Intuitively, $\mathcal{E}_{Bob}$ is a set of Bob's secrets. Bob can use these secrets for secret random values and for secret messages. With Alice's public key in T, Bob creates an encrypted packet M_{Bob}. Note that M_{Bob} doesn't have to be exactly one message, with iteration of function symbol "$\langle , \rangle$", Bob can deliver a number of messages. The sets $T_{Alice} \cup \{M_{Bob}\}$ and $T_{Cust} \cup \{M_{Bob}\}$ model that Alice and her Custodians have received the packet M_{Bob}. When $(\mathcal{E} \cup \mathcal{E}_{Bob}, T_{Alice} \cup \{M_{Bob}\}) \vdash m$ holds, it means that Alice has successfully decrypted the packet M_{Bob} and obtain a secret m from Bob. The condition $m \in \mathcal{E}_{Bob}$ rules out some trivial cases. For example, Alice can derive her own secret key, i.e. $(\mathcal{E} \cup \mathcal{E}_{Bob}, \Gamma_{Alice} \cup \{M_{Bob}\}) \vdash a$, but clearly her Custodians cannot, i.e. $(\mathcal{E} \cup \mathcal{E}_{Bob}, \Gamma_{Cust} \cup \{M_{Bob}\}) \nvdash a$.

The above Theorem 8 states a non-interactive property. We do not consider the interactive attacks which cannot usually be prevented by software escrow. Even for the key escrow systems [37], [47], [43], users can always abuse the facilities if they collaborate to do so. For example, Alice could directly send a new public key to Bob, or Alice could abuse the certificate of her public key to sign and certify a new public key, or they could perform a Diffie-Hellman key exchange to establish a secret session key, or they could establish a shared secret key by meeting each other physically. Our system is not intended to provide censorship and it cannot stop Alice and Bob from communicating privately by other means. After all, if users want to create their own private channels, they can always use OpenPGP or S/MIME.

4.3 Proof of Theorem 8

The main idea of proving Theorem 8 is to transform every derivation $(\mathcal{E} \cup \mathcal{E}_{Bob}, T_{Alice} \cup \{M_{Bob}\}) \vdash m$ by Alice into a derivation by custodians by swapping the names a, c. However, because the set T is not completely symmetric w.r.t. Alice and Custodians, i.e. $T\{a \mapsto c, c \mapsto a\} \neq T$, thus $T_{Alice}\{a \mapsto c, c \mapsto a\} \neq T_{Cust}$. Hence we need to protect the c in g^c before swapping a, c.

We choose two different fresh names n_1, n_2. We construct a substitution $\sigma_{sc} := \{c \mapsto n_1 \mid$ there exists U such that c occurred in U and $U =_{AC} s \cdot c \cdot X$ for some $X\}$ and $\sigma_{sc}^{-1} := \{n_1 \mapsto c\}$, and $\sigma_c := \{c \mapsto n_2\}$ and $\sigma_c^{-1} := \{n_2 \mapsto c\}$. Let $\sigma_{ac} := \{a \mapsto c, c \mapsto a\}$. We construct a substitution σ_{sac} by combining all the substitutions together:

$$\sigma_{sac} := \sigma_{sc}\, \sigma_c\, \sigma_{sc}^{-1}\, \sigma_{ac}\, \sigma_c^{-1}$$

The main purpose of σ_{sc} and σ_c is to distinguish the c in g^{sc} and c in g^c.

Given $(\mathcal{E} \cup \mathcal{E}_{Bob}, T_{Alice} \cup \{M_{Bob}\}) \vdash m$, we shall prove that $(\mathcal{E}\sigma_{sac} \cup \mathcal{E}_{Bob}\,\sigma_{sac},\ T_{Alice}\sigma_{sac} \cup \{M_{Bob}\,\sigma_{sac}\}) \vdash m\sigma_{sac}$ is a valid derivation. Note that $(\mathcal{E}\sigma_{sac} \cup \mathcal{E}_{Bob}\sigma_{sac}, T_{Alice}\sigma_{sac} \cup \{M_{Bob}\,\sigma_{sac}\}) = (\mathcal{E} \cup \mathcal{E}_{Bob}, T_{Cust} \cup \{M_{Bob}\})$ and also $m\sigma_{sac} = m$. So if derivation relation $\vdash$ after substitution is proved, it will give us the conclusion.

To make our reasoning easier, we need to orient the rules in the equations $E_{DH} \cup E_{Crypt}$. The first equivalent orientation of Abelian Group was proposed by Lankford [35]. Here we use the automatic rewriting tool CiME [18] to obtain an equivalent orientation. The tool generates the following auxiliary rules R_{Aux}:

$$\begin{array}{lcllcl} (x \cdot y)^{-1} \cdot y & \to & x^{-1} & x \cdot (x^{-1} \cdot y) & \to & y \\ (x^{-1} \cdot y)^{-1} & \to & x \cdot y^{-1} & (x^{-1})^y \cdot x^y & \to & 1 \\ ((x^{-1})^y)^{-1} & \to & x^y & & & \end{array}$$

Let R_{AG} be a set of rewriting rules obtained by orienting from left to right the equations E_{AG} of Abelian group except the AC rules. Let R_{Exp} be a rewriting system obtained by orienting the exponentiation rules E_{Exp} in Def. 5 from left to right, and also R_{Crypt} by orienting the equations in E_{Crypt} from left to right.

Lemma 9. *Let $R_{DHC} = R_{AG} \cup R_{Exp} \cup R_{Aux} \cup R_{Crypt}$. R_{DHC} is AC-convergent.*

Proof: This lemma is proved by the tool CiME [18]. ■

We define $\vdash'$ as exactly the same as $\vdash$ except that $=_{DHC}$ is replaced with $\xrightarrow{*}_{R_{DHC}/AC}$. Similarly to Proposition 7, we can see that:

Proposition 10. *$(\mathcal{E}, T) \vdash' L$ iff there exists a context $\mathcal{C}$ with $name(\mathcal{C}) \cap \mathcal{E} = \emptyset$ and a set of terms $L_1, \ldots, L_k \in T$ such that $\mathcal{C}[L_1, \ldots, L_k] \xrightarrow{*}_{R_{DHC}/AC} L$.*

By Proposition 7 and the hypothesis $(\mathcal{E} \cup \mathcal{E}_{Bob}, T_{Alice} \cup \{M_{Bob}\}) \vdash m$, there exists $\widehat{\mathcal{C}}[N_1, \ldots, N_i]$ with $name(\widehat{\mathcal{C}}) \cap (\mathcal{E} \cup \mathcal{E}_{Bob}) = \emptyset$ and $N_1, \ldots, N_i \in T_{Alice}$ such that $\widehat{C}[N_1, \ldots, N_i] =_{DHC} m$. Since m is a name and there is no rule in $R_{DH} \cup R_{Crypt}$ for a name, by AC-confluence of R_{DHC} stated in Lemma 9, we have $\widehat{\mathcal{C}}[N_1, \ldots, N_i] \xrightarrow{*}_{R_{DHC}/AC} m$. Hence we can see that

$$(\mathcal{E} \cup \mathcal{E}_{Bob}, T_{Alice} \cup \{M_{Bob}\}) \vdash' m$$

Now we proceed to substitute the above derivation sequence by our σ_{sac}. We need the following two claims:

Claim 1: for any $(\mathcal{E} \cup \mathcal{E}_{Bob}, T_{Alice} \cup \{M_{Bob}\}) \vdash' L$, if $L = \mathcal{D}[s]$ (also $\mathcal{D}[c]$) for some context $\mathcal{D}$, then there exist a context $\widehat{\mathcal{D}}$ and terms X_1, X_2 such that $\mathcal{D} = \widehat{\mathcal{D}}[g^{X_1[\cdot]X_2}]$ and $s \notin name(X_1, X_2)$ (resp. $c \notin name(X_1, X_2)$).

Claim 1 states that s (also c) can only occur in a form $g^{X_1\, s\, X_2}$ (resp. $g^{X_1\, c\, X_2}$) in any derivation made by Alice.

Claim 2: if $(\mathcal{E} \cup \mathcal{E}_{Bob}, T_{Alice} \cup \{M_{Bob}\}) \vdash' L$, then $(\mathcal{E}\sigma_{sac} \cup \mathcal{E}_{Bob}\sigma_{sac}, T_{Alice}\sigma_{sac} \cup \{M_{Bob}\sigma_{sac}\}) \vdash' L\sigma_{sac}$.

Proof: The proof proceeds by induction on the derivation of $\vdash'$. Assume $(\mathcal{E} \cup \mathcal{E}_{Bob}, \Gamma_{Alice} \cup \{M_{Bob}\}) \vdash' L'$ and $L' \rightarrow_{R_{DHC}/AC} L$. We take the rule $(x \cdot y)^{-1} \cdot y \rightarrow x^{-1}$ as an example and the other cases are similar. Assume there exists a context $\mathcal{C}$ and a term U and a substitution σ such that $L =_{AC} \mathcal{C}[U]$ and $U = ((x \cdot y)^{-1} \cdot y)\sigma$ and $L' =_{AC} \mathcal{C}[x^{-1}\sigma]$. The interesting case is when $y\sigma = s$ and s is eliminated by the rewriting rule which may result in $\mathcal{C}$ and $x\sigma$ being substituted by σ_{sac} in different ways in L and L'. However, this is impossible because s can only occur in a form $g^{X_1 \cdot s \cdot X_2}$ according to Claim 1. ■

Using Claim 2 and the fact that $(\mathcal{E} \cup \mathcal{E}_{Bob}, T_{Alice} \cup \{M_{Bob}\}) \vdash' m$, we can get $(\mathcal{E}\sigma_{sac} \cup \mathcal{E}_{Bob}\,\sigma_{sac},\ T_{Alice}\sigma_{sac} \cup \{M_{Bob}\,\sigma_{sac}\}) \vdash' m\sigma_{sac}$. From $(\mathcal{E}\sigma_{sac} \cup \mathcal{E}_{Bob}\sigma_{sac}, T_{Alice}\sigma_{sac} \cup \{M_{Bob}\,\sigma_{sac}\}) = (\mathcal{E} \cup \mathcal{E}_{Bob}, T_{Cust} \cup \{M_{Bob}\})$ and the definition of $\vdash$, it holds that $(\mathcal{E} \cup \mathcal{E}_{Bob}, T_{Cust} \cup \{M_{Bob}\}) \vdash m$.

It is worth pointing out that without orienting equations of $=_{DHC}$ to R_{DHC}, the substitution σ_{sac} would be difficult to be applied on equations in $=_{DHC}$. For example, the rule $\mathsf{sdec}(\mathsf{senc}(x, y, z), y) = x$ would rewrite name s into $\mathsf{sdec}(\mathsf{senc}(s, s', s''), s')$ for some arbitrary names s', s'', and the rule $(u^x)^y$ would split names s and c.

5 Accountability granularity

Often, it might not be desirable to allow citizens to have the access to the details of an investigation. This is because, for example, it could alert criminals to the fact of an investigation concerning them. Despite this, we envisage there being rules, decided democratically, for what sort of access to the investigation is allowed. We can imagine a spectrum starting with a version that is "generous" to the citizen, by providing maximal access, with various degrees of generosity and ending with a "mean" version:

- The generous version gives a full account, for a given individual, of all the accesses to their data that have been investigated.
- An intermediate version gives a time-delayed and quantitative account, but it lacks detail. For example, the individual gets to read the proportion of journeys whose data has been accessed that have taken place more than, say, two years ago.
- A mean version gives a real-time but much coarser view. An individual can read the proportion of accesses made for all journeys taken by everyone, but cannot see which accesses are about her.

While the first one can be set as a long-term goal for democracy, the third one meets the current appeal for government surveillance reform for quantity and nature of government demands to be published [17]. We focus on the third type of accountability in this paper. Our log is public and citizens get to monitor the quantity of government demands in real time, while the confidentiality of investigations is protected.

6 Future work

The paper so far has focused on asynchronous person-to-person communication. The concept of balancing societal security and individual privacy can be applied to many other areas. We make brief remarks about some of them.

Wireless tickets: Wireless ticket systems (such as the London Oyster card, the Paris Navigo card, or Washington's SmarTrip card) allow passengers to travel on city-wide transport by presenting a contactless smartcard at the time of taking a journey. With a wireless ticket, a passenger's journeys are logged and stored in perpetuity. To combat terrorism, and to avoid the need to obtain court orders each time, the UK intelligence agencies MI5 and MI6 have sought full automated access to Transport for London's Oyster smart card database. The data could potentially be used not just for law enforcement but potentially for advertising purposes, or even criminal stalking and harassment.

We are working on designing a system which would allow passengers to encrypt their journey information and would also allow law enforcement agencies to search and decrypt the data while being held accountable.

Digital cash: The online crypto currency Bitcoin allows one to receive cash merely by providing a public key, and to spend it by creating a signature with the corresponding private key. A user can create as many keys as she likes, and can use Tor or other means to hide her IP address. In this way, Bitcoin offers high degrees of privacy, and therefore has become an instrument for illegal traders and money launderers [5]. Recently, law enforcers have begun making arrests [4], [13] and possibly shutting down exchanges [3], and as a result it is possible that Bitcoin will fail completely. A digital currency with better privacy/security balance might be more attractive to society.

Communications metadata: Metadata are the data providing information about a communication, such as email headers, phone numbers, length of communication, and

IP address and cookies from websites. The retention of metadata is currently mandatory by law in many countries. Law enforcers can obtain access to details of who one has texted, the location from which a text was sent or received, and the dates and times of text messages. Preventing service providers having access to metadata is technically difficult, because they need it to provide the service.

7 Conclusions

We propose a general idea called "accountable escrow" for balancing the conflicting requirements of societal security and individual privacy. We design a balanced crypto system for asynchronous communication, that uses a novel method for escrowing decryption capability for public key cryptography. Our system allows citizens to encrypt their private information, which prevents snooping from service providers, foreign governments, and other third party attackers, while also allows the domestic government to perform decryptions in order to conduct investigations. Those governments are held accountable for the nature and quantity of decryptions. Citizens can monitor the quantity of government demands in real time.

References

[1] Academics Against Mass Surveillance. http://www.academicsagainstsurveillance.net/.

[2] BBC 5 Live. George Galloway MP says Edward Snowden deserves a medal. https://twitter.com/bbc5live/status/345300092300976128.

[3] Catherine Shu. MtGox Temporarily Pauses Bitcoin Withdrawals. 6 February 2014. http://techcrunch.com/2014/02/06/mt-gox-temporarily-pauses-bitcoin-withdrawals/.

[4] Dave Lee. US makes Bitcoin exchange arrests after Silk Road closure BBC News, 28 January 2014. http://www.bbc.co.uk/news/technology-25919482.

[5] Emily Flitter. FBI shuts alleged online drug marketplace, Silk Road. Reuters, 2 October 2013. http://news.yahoo.com/fbi-raids-alleged-online-drug-market-silk-road-153729457.html.

[6] Fifty-three prominent US academics. *An Open Letter from US Researchers in Cryptography and Information Security*. 24 January 2014. http://masssurveillance.info/.

[7] International Business Times. GCHQ Forced Secure Email Service PrivateSky to Shut Down. 11 December 2013. http://www.ibtimes.co.uk/gchq-forced-privatesky-secure-email-service-offline-529392.

[8] Kenneth Paterson, Mark Ryan, Peter Ryan, Vladimiro Sassone, Steve Schneider, Nigel P. Smart, Eerke Boiten, George Danezis, Jens Groth and Feng Hao. *Open Letter From UK Security Researchers.* 16 September 2013. http://bristolcrypto.blogspot.co.uk/2013/09/open-letter-from-uk-security-researchers.html.

[9] Phil Rogaway. A cryptographer's Statement of condemnation of U.S. mass-surveillance programs, and a reminder of our ethical responsibilities as computer scientists. http://www.cs.ucdavis.edu/~rogaway/.

[10] Reform Government Surveillance. http://reformgovernmentsurveillance.com/.

[11] The Guardian. *Edward Snowden voted Guardian person of the year 2013*. http://www.theguardian.com/world/2013/dec/09/edward-snowden-voted-guardian-person-of-year-2013.

[12] The Guardian. Lavabit email service abruptly shut down citing government interference. 9 August 2013. http://www.theguardian.com/technology/2013/aug/08/lavabit-email-shut-down-edward-snowden.

[13] The Guardian. Silk Road's alleged frontman indicted. 5 February 2014. http://www.theguardian.com/technology/2014/feb/05/silk-roads-alleged-founder-indicted.

[14] The Register. Lavabit founder: Feds ORDERED email providers to stay open. 31 October 2013. http://www.theregister.co.uk/2013/11/19/lavabit_analysis/.

[15] Forbes. Encryption App Silent Circle Shuts Down E-Mail Service 'To Prevent Spying'. 8 September 2013. http://www.forbes.com/sites/parmyolson/2013/08/09/encryption-app-silent-circle-shuts-down-e-mail-service-to-prevent-spying/.

[16] PrivateSky - Secure email and file transfer encryption service. http://www.cloudoneservices.com/privatesky/.

[17] Reform Government Surveillance. http://www.reformgovernmentsurveillance.com/.

[18] The CiME Rewrite Tool. http://cime.lri.fr/.

[19] The real story on the PrivateSky takedown. 12 December 2013. http://www.certivox.com/blog/bid/359788/The-real-story-on-the-PrivateSky-takedown.

[20] Certificate transparency. Available: www.certificate-transparency.org, 2007.

[21] M. Abadi and C. Fournet. Mobile values, new names, and secure communication. In *Proc. 28th Symposium on Principles of Programming Languages (POPL'01)*, pages 104–115. ACM Press, 2001.

[22] M. Abdalla, M. Bellare, and P. Rogaway. The Oracle Diffie-Hellman Assumptions and an Analysis of DHIES. In *CT-RSA*, pages 143–158, 2001.

[23] H. Abelson, R. Anderson, S. M. Bellovin, J. Benaloh, M. Blaze, W. Diffie, J. Gilmore, P. G. Neumann, R. L. Rivest, J. I. Schiller, and B. Schneier. The Risks of Key Recovery, Key Escrow, and Trusted Third-Party Encryption. https://www.schneier.com/paper-key-escrow.html.

[24] M. Bellare and S. Goldwasser. Verifiable partial key escrow. In *Proc. 4th ACM Conference on Computer and Communications Security*, 1997.

[25] M. Blaze. Protocol failure in the escrowed encryption standard. In *Proceedings of the 2Nd ACM Conference on Computer and Communications Security*, CCS '94, pages 59–67, New York, NY, USA, 1994. ACM.

[26] D. Boneh and M. K. Franklin. Identity-based encryption from the weil pairing. CRYPTO '01, pages 213–229. Springer-Verlag, 2001.

[27] F. Brandt. Efficient cryptographic protocol design based on distributed El Gamal encryption. In *ICISC*, pages 32–47. SpringerVerlag, 2005.

[28] E. Bresson, O. Chevassut, and D. Pointcheval. The Group Diffie-Hellman Problems. In *Selected Areas in Cryptography*, pages 325–338, 2002.

[29] S. Bursuc, G. S. Grewal, and M. D. Ryan. Trivitas: Voters directly verifying votes. In *VOTE-ID*, pages 190–207, 2011.

[30] D. Chaum and T. P. Pedersen. Wallet databases with observers. In *Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '92, pages 89–105, 1993.

[31] H. Comon-Lundh and S. Delaune. The finite variant property: How to get rid of some algebraic properties. In *RTA*, pages 294–307, 2005.

[32] T. El Gamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In *Proceedings of CRYPTO 84 on Advances in Cryptology*, pages 10–18, 1985.

[33] E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Advances in Cryptology - CRYPTO '99*, volume 1666, pages 537–554, 1999.

[34] G. S. Grewal, M. D. Ryan, S. Bursuc, and P. Y. A. Ryan. Caveat coercitor: Coercion-evidence in electronic voting. In *IEEE Symposium on Security and Privacy*, pages 367–381, 2013.

[35] J.-M.Hullot. A catalogue of canonial term rewriting systems. Computer Science Laboratory, SRI, CA,USA,1980., 1980. Technical Report, CSL-114.

[36] A. Joux. A one round protocol for tripartite diffie-hellman. In *Proceedings of the 4th International Symposium on Algorithmic Number Theory*, ANTS-IV, pages 385–394, London, UK, UK, 2000. Springer-Verlag.

[37] J. Kilian and F. T. Leighton. Fair cryptosystems, revisited: A rigorous approach to key-escrow (extended abstract). In *Proceedings of the 15th Annual International Cryptology Conference on Advances in Cryptology*, CRYPTO '95, pages 208–221. Springer-Verlag, 1995.

[38] J. Kilian and E. Petrank. Identity escrow. In *In CRYPTO '98*, pages 169–185. Springer-Verlag, 1997.

[39] S. Kremer, M. Ryan, and B. Smyth. Election verifiability in electronic voting protocols. In *ESORICS*, pages 389–404, 2010.

[40] K. Kurosawa and T. Matsuo. How to remove mac from dhies. In *ACISP*, pages 236–247, 2004.

[41] Mark D. Ryan. Enhanced certificate transparency and end-to-end encrypted mail. In Network and Distributed System Security (NDSS), 2014.

[42] R. C. Merkle. A digital signature based on a conventional encryption function. CRYPTO '87, pages 369–378, 1988.

[43] P. Paillier and M. Yung. Self-escrowed public-key infrastructures. In *ICISC*, pages 257–268, 1999.

[44] T. P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. CRYPTO '91, pages 129–140, 1992.

[45] M. Steiner, G. Tsudik, and M. Waidner. Diffie-hellman key distribution extended to group communication. CCS '96, pages 31–37. ACM, 1996.

[46] A. Whitten and J. D. Tygar. Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0. In *Proceedings of the 8th Conference on USENIX Security Symposium - Volume 8*, SSYM'99, pages 14–14, 1999.

[47] A. L. Young and M. Yung. Auto-recoverable auto-certifiable cryptosystems. In *EUROCRYPT*, volume 1403 of *Lecture Notes in Computer Science*, pages 17–31. Springer, 1998.