Semantically Secure Anonymity: Foundations of Re-Encryption

Adam L. Young
Cryptovirology Labs

Moti Yung
Snapchat and Dept. of Computer Science,
Columbia University

ABSTRACT

The notion of universal re-encryption is an established primitive used in the design of many anonymity protocols. It allows anyone to randomize a ciphertext without changing its size, without decrypting it, and without knowing the receiver's public key. By design it prevents the randomized ciphertext from being correlated with the original ciphertext. We revisit and analyze the security foundation of universal re-encryption and show that to date it has not had a satisfactory definition of security, in spite of its numerous uses. We then analyze the anonymity arguments for the ElGamal-based universal cryptosystem and show that it has not been proven to be anonymous under DDH (and does not meet the standards of modern cryptography), and that such a proof is non-trivial given existing reduction techniques. This analysis is a type of cryptanalysis of provably secure systems, where reductions and exact assumptions have certain gaps in them that need to be detected and corrected. The notion of an incomparable public key cryptosystem is closely related to universal re-encryption; we similarly cryptanalyze the security foundation of the ElGamal-based incomparable public key cryptosystem as well and show that it was not proven to be secure. To correct the lack of foundation, we introduce a definition of what properties are needed for a re-encryption cryptosystem that needs to provide anonymity. We then introduce a new generalization of the well-known Decision Diffie-Hellman (DDH) random self-reduction and use it, in turn, to prove that the ElGamal-based universal cryptosystem is secure under DDH. We apply our new DDH reduction technique to incomparable public key cryptosystems as well and prove that it is secure, and, as a new application, we present a novel secure *Forward-Anonymous Batch Mix*.

Keywords

probabilistic re-encryption, end-to-end security, key anonymity, anonymous communication, semantic security, message indistinguishability, batch mix, DDH groups, cryptanalysis of provably secure cryptosystems.

1. INTRODUCTION

Privacy and anonymity tools have become increasingly important for maintaining basic civil liberties. For example, as a result of the whistle-blowing by Edward Snowden, Americans and others have a better understanding of surveillance states and the privacy risks they pose. This is a reminder of the importance of anonymity of communication, which, in fact, has been an active area of cryptographic research since the 1980's. Having a sound theoretical foundation for anonymity systems is a critical component in achieving privacy of users in the same way that message security is achieved by having a sound theoretical foundation for encryption. Camenisch and Lysyanskaya, for example, presented a formal treatment of onion routing [3], where prior work was comparatively informal with ad-hoc security justifications. Onion routing falls into a class of anonymity systems known as "decryption mixes", since layers of ciphertext are shed as the onion makes its way to the receiver.

In this work we present a formal treatment of a different fundamental class of anonymous communication protocols, namely, those based on universal re-encryption. This concept forms the basis of what has been called "re-encryption mixes". Golle, Jakobsson, Juels, and Syverson introduced the notion of universal re-encryption and presented a cryptosystem that implements it called `UCS` [9]. A ciphertext in a re-encryption mix has the property that it can be efficiently re-encrypted by anyone without knowledge of the receiver's public key. This is accomplished without ever exposing the underlying plaintext and without changing the size of the ciphertext. Using re-encryption randomness, the mapping is "lost" between the ciphertext that is supplied to the re-encryption operation and the resulting output ciphertext. Therefore, the notion of universal re-encryption propelled anonymity into the important area of "end-to-end encryption" systems that do not rely on servers for maintaining secrecy and that have the forward-secrecy property. Forward-secrecy and end-to-end encryption are becoming increasingly important in industrial systems in the post-Snowden era.

A number of anonymity protocols have been constructed that utilize universal re-encryption. Here, we show by way of cryptanalyzing the associated security definitions and proofs (i.e., a cryptanalysis methodology applied to provably secure systems) that the required level of security of this cryptographic primitive has not been shown. We show by way of examples that the previous security definitions failed to properly capture the anonymity property. We go on to show

that the previous security proofs have not been tied to the hardness of Decision Diffie-Hellman (DDH) contrary to what has been claimed in the literature.

To show the message semantic security of re-encryption the usual approach of exploiting random self-reducibility of exponentiation, say, suffices (a requirement is to maintain the ability to decrypt correctly). For this property it is easy to show that semantic security is retained. However, in the case of anonymity where there is a complex goal that includes semantic security [7], key-privacy [1] (essential to hiding the receiver in an end-to-end fashion), and indistinguishability between pre and post re-encryption (essential for sender-receiver unlinkability), the needed properties and formal notions of security for this setting were neglected and were, in fact, never done.

As a result of this missing foundational step, all claims for use of re-encryption in anonymity systems have been based on heuristics. Indeed, there are no properties required, nor proofs that they are achieved, in the papers exploiting re-encryption for anonymity.

What is needed is a formal foundation of the field (as was done in other areas, e.g. message encryption). To this end, we put forth a model of what is required for re-encryption in the context of anonymity systems and show that the existing constructions do not lead to proofs of these properties. Therefore, a new procedure for re-encryption in anonymity systems is needed as well. The hope is that the model, definitions, and construction will initiate a more foundational approach to using cryptography within anonymity systems in the tradition of modern cryptography (building on the early works, indeed). We point out that our investigation, in fact, follows the traditional methodology of modern cryptography where similar corrections of definitions, constructions, proofs, and reductions have been a central theme aimed at establishing proper primitives to build security and privacy applications and protocols thereupon.

Our analysis shows an unjustified trend towards claiming security with respect to DDH but not formally proving it. Note that maintaining message semantic security under re-encryption is relatively trivial. But, this is misleading since it does not hold for the entire set of properties needed for anonymity applications. These definitions have never been given before. In particular, "anonymity" of the probabilistic encryption and re-encryption algorithms were not properly defined in *any* prior work. As we know from other areas of cryptography, without foundations following a model, definitions, constructions and careful proofs, issues will surely arise.

We summarize our contributions as follows:

1. Modeling: We cryptanalyze the definition of a universal cryptosystem and show that it is not sufficient. We define what we call *semantically secure anonymity* that defines the (complete set of) security properties that assure anonymity, that existing protocols take for granted.

2. Cryptanalysis and criticism: We cryptanalyze the security proofs given for the ElGamal-based universal cryptosystem and show that it has not been proven secure under DDH. We show that proving that key anonymity [1] holds is non-trivial.

3. Construction: We generalize the well-known DDH random self-reduction and then use this generalization to prove that the ElGamal-based universal cryptosystem is secure under DDH as modeled. This is a new reduction technique that may have independent applications.

4. A notion very closely related to universal re-encryption is that of *incomparable public keys*. We cryptanalyze the proof of security for the ElGamal-based incomparable public key cryptosystem from ACM CCS 2003 [21] and show that it has not been proven secure under DDH, contrary to what was claimed. We then apply our reduction technique to give the first proof that incomparability holds under DDH.

5. Application to New Protocols: As an application we present a forward-anonymous batch mix that is secure (as modeled in here) under DDH.

We anticipate that our new reduction technique will aid in future concrete and workable designs that use number theoretic and elliptic curve groups where DDH holds, since anonymity of channels is a central issue in cryptography and privacy applications and since sound foundations and correct proofs are needed. Finally, our new application of a forward-anonymous batch mix that we prove secure is an example of such an application that gives an end-to-end secure anonymous communication system.

Organization: In Section 2 we present related work. Notation and definitions are covered in Section 3. We review universal re-encryption (`UCS`) in Section 4 and cryptanalyze it. The new DDH reduction technique is covered in Section 5 and it is used to prove the security of universal re-encryption in Section 6. We cryptanalyze the ElGamal-based incomparable public key cryptosystem in Section 7 and then repair its security foundation. Our forward-secure batch mix is covered in Section 8 and we prove that it is secure in Section 9. We present our conclusions in section 10.

2. RELATED WORK

Let us first review the literature that leverages universal re-encryption as a primitive.

Jakobsson et al presented a universal re-encryption cryptosystem that they referred to as UCS [9]. UCS is a 4-tuple of algorithms: a key generator, encryption algorithm, re-encryption algorithm, and a decryption algorithm. It is an extension of the ElGamal public key cryptosystem [5]. A ciphertext produced using this cryptosystem can be re-encrypted by anyone without first deciphering it. They present two applications that leverage a universal cryptosystem. In the first application an RFID tag is set to be a universal ciphertext that contains an underlying ID as the plaintext. The ciphertext is re-randomized periodically to prevent the tag from being tracked over time, e.g., as the object that contains the tag moves from place to place. With the private decryption key the ID can be obtained. Without the private key the ID in the ever changing RFID ciphertext cannot be obtained, making it difficult to track the object. They also apply universal re-encryption to construct a hybrid universal mix that leverages a public bulletin board. The mix is based on uploading and downloading ciphertexts to/from a bulletin board as opposed to leveraging a cascade of mix servers.

Fairbrother sought a more efficient hybrid universal cryptosystem based on UCS [6]. Universal re-encryption was used in a protocol to control anonymous information flow, e.g., to prevent spam from being injected into the anonymization network [13]. Onion-based routing and universal re-encryption were leveraged to form hybrid anonymous communication protocols [10, 14]. A circuit-based anonymity protocol was presented based on universal re-encryption [15]: in the first stage a channel is established through the network between Alice and Bob along with the keys needed for re-encryption and in the second stage Alice and Bob communicate with one another. Weakness in [13, 14, 10, 15] were presented in [4]. Golle presented a *reputable mix network* construction based on universal re-encryption [8]. A reputable mix has the property that the mix operator can prove that he or she did not author the content output by the mix.

A concept that is closely related to a universal cryptosystem is an *incomparable public key cryptosystem* [21]. An incomparable public key cryptosystem has the property that senders are not able to determine who the receivers are. An incomparable public key is an encryption of unity under a traditional ElGamal public key. The incomparable public key can be used to re-randomize itself to form equivalent public keys. A universal ciphertext is equivalent to a ciphertext from the incomparable public key cryptosystem along with the incomparable public key of the receiver.

Groth presented a re-randomizable and replayable adaptive chosen ciphertext attack secure cryptosystem based on DDH [11]. The construction and security arguments do not address key-anonymity.

Prabhakaran and Rosulek presented a construction for a rerandomizable encryption scheme [18] that aims to be CCA-secure under DDH. It extends the Cramer-Shoup public key cryptosystem. They define RCCA receiver-anonymity in detail but state that their scheme does not achieve it and that it is an open problem. The approach was later extended to combine computability features with non-malleability of ciphertexts. The construction enables anyone to change an encryption of an unknown message m into an encryption of $T(m)$ (a feature), for a set of specific allowed functions T, but is non-malleable with respect to all other operations [19]. They indicate that their construction does not achieve HCCA-anonymity and leave the anonymity problem as open.

There has been recent work on proxy encryption [12]. In proxy encryption a ciphertext of a message m encrypted under Alice's public key is transformed (re-encrypted) into a ciphertext of m under Bob's public key. Note that our setting is different since the receiver's public key does not change in our re-encryption operation.

Having surveyed the literature it became apparent to us that numerous works have utilized universal re-encryption as a basic building block. This forms the motivation for a clean and correct foundation for this area. While we fully appreciate the pioneering work on this concept (a trailblazing step which is necessary), we believe that the time has come to treat anonymity with the same formal care and level of provability as, say, message security in public key cryptosystems. We believe that our work shows that identifying subtleties and producing necessary revisions is relevant, even for works that are older than 10 years, especially in areas that are becoming increasingly important to real-world applications.

The notion of key privacy (also called key anonymity) was introduced by Boldyreva et al [1]. They formally defined public key cryptosystems that produce ciphertexts that do not reveal the receiver and showed that ElGamal and Cramer-Shoup achieve key privacy.

3. NOTATION AND DEFINITIONS

If T is a finite set then $x \in_U T$ denotes sampling x uniformly at random from T. Define $\mathbb{Z}_p$ to be $\{0, 1, 2, ..., p-1\}$. Let $\mathbb{Z}_n^*$ be the set of integers from $\mathbb{Z}_n$ that are relatively prime to n. $[1, t]$ denotes the set of integers $\{1, 2, ..., t\}$. $|\mathbb{G}|$ denotes the size of the group $\mathbb{G}$, i.e., number of elements in $\mathbb{G}$. We may omit writing "mod p" when reduction modulo p is clear from the context. $\Pr[A]$ denotes the probability that A is true. Let $a \leftarrow b$ denote the assignment of b to a. For example, $a \leftarrow M(x)$ denotes the execution of Turing machine M on input x resulting in output a.

The following definition of DDH is directly from [2]. A group family $\mathbb{G}$ is a set of finite cyclic groups $\mathbb{G} = \{G_{\mathfrak{p}}\}$ where $\mathfrak{p}$ ranges over an infinite index set. We denote by $|\mathfrak{p}|$ the size of the binary representation of $\mathfrak{p}$. We assume that there is a polynomial time (in $|\mathfrak{p}|$) algorithm that given $\mathfrak{p}$ and two elements in $G_{\mathfrak{p}}$ outputs their sum. An instance generator, $\mathcal{IG}$, for $\mathbb{G}$ is a randomized algorithm that given an integer n (in unary), runs in time polynomial in n and outputs some random index $\mathfrak{p}$ and a generator g of $\mathbb{G}_{\mathfrak{p}}$. In particular, $(\mathfrak{p}, g) \leftarrow \mathcal{IG}(n)$. Note that for each n, the instance generator induces a distribution on the set of indices $\mathfrak{p}$. The index $\mathfrak{p}$ encodes the group parameters.

A DDH algorithm $\mathcal{A}$ for $\mathbb{G}$ is a probabilistic polynomial time Turing machine satisfying, for some fixed $\alpha > 0$ and sufficiently large n:

$$|\Pr[\mathcal{A}(\mathfrak{p}, g, g^a, g^b, g^{ab}) = \text{“true”}] - \Pr[\mathcal{A}(\mathfrak{p}, g, g^a, g^b, g^c) = \text{“true”}]| > \frac{1}{n^\alpha}$$

where g is a generator of $G_{\mathfrak{p}}$. The probability is over the random choice of $\langle \mathfrak{p}, g \rangle$ according to the distribution induced by $\mathcal{IG}(n)$, the random choice of a, b, and c in the range $[1, |G_{\mathfrak{p}}|]$ and the random bits used by $\mathcal{A}$. The group family $\mathbb{G}$ satisfies the DDH assumption if there is no DDH algorithm for $\mathbb{G}$.

We now review the well-known random-self reduction for DDH [2, 20, 17]. DDHRerand$((p, q), g, x, y, z)$ randomizes a DDH problem instance by choosing $u_1, u_2, v \in_U [1, q]$ and computing,

$$(x', y', z') \leftarrow (x^v g^{u_1}, y g^{u_2}, z^v y^{u_1} x^{vu_2} g^{u_1 u_2})$$

When (x, y, z) is a valid Diffie-Hellman 3-tuple then the output is a random Diffie-Hellman 3-tuple. When (x, y, z) is not a valid Diffie-Hellman 3-tuple then the output is a random 3-tuple.

4. CRYPTANALYSIS OF UCS

4.1 Review of universal re-encryption

Let k be a security parameter and let $\mathfrak{p} = (p, q)$ be a group family where p is prime and $p - 1$ is divisible by a large prime q. The group $G_{\mathfrak{p}}$ is the subgroup of $\mathbb{Z}_p^*$ having order q. For anonymity, the single group $((p, q), g)$ is generated once using $\mathcal{IG}(k)$ and is then used by all users.

Key Generation: Key generation is denoted by $(y, x) \leftarrow \texttt{UGEN}((p, q), g)$. Here $y \leftarrow g^x \bmod p$ where $x \in_U [1, q]$. The public key is $(y, g, (p, q))$ and the private key is x.

Encryption: The following encryption operation is denoted by,

$$\texttt{UENCR}(m, (k_0, k_1), (y, g, p))$$

It encrypts message $m \in G_{\mathfrak{p}}$ using y. $(k_0, k_1) \in_U [1, q] \times [1, q]$ are random encryption nonces. The operation outputs the ciphertext $c \leftarrow ((a_0, b_0), (a_1, b_1)) \leftarrow ((g^{k_0} \bmod p, y^{k_0} \bmod p), (g^{k_1} \bmod p, y^{k_1} m \bmod p))$.

Universal Re-encryption: The following is the universal re-encryption operation,

$$\texttt{URENC}(((a_0, b_0), (a_1, b_1)), (\ell_0, \ell_1), p)$$

The pair $((a_0, b_0), (a_1, b_1))$ is a universal ciphertext produced using UENCR and the value $(\ell_0, \ell_1) \in_U [1, q] \times [1, q]$ is a pair of re-encryption nonces. Compute $(\alpha_0, \beta_0) \leftarrow (a_0^{\ell_0} \bmod p, b_0^{\ell_0} \bmod p)$ and compute $(\alpha_1, \beta_1) \leftarrow (a_1 a_0^{\ell_1} \bmod p, b_1 b_0^{\ell_1} \bmod p)$. Output the ciphertext $c \leftarrow ((\alpha_0, \beta_0), (\alpha_1, \beta_1))$. We supply p as an argument whereas Golle et al do not (an extremely minor oversight).

Decryption: The following decryption operation is denoted by $\texttt{UDECR}(c, x, p)$. Here c is the ciphertext $((a_0, b_0), (a_1, b_1))$. Compute $m_0 \leftarrow b_0 / a_0^x \bmod p$. If $m_0 = 1$ then set $s =$ true else set $s =$ false. If $s =$ true set $m_1 = b_1 / a_1^x \bmod p$ else set m_1 to be the empty string. $s =$ true indicates successful decryption. Return (m_1, s).

The encryption nonces are parameterized to facilitate our proofs of security. The UCS cryptosystem is the 4-tuple (UGEN, UENCR, URENC, UDECR).

4.2 The Need for Anonymity Definitions

Key anonymity needs to be defined and proven for the universal encryption operation. We show this by way of example. Suppose that Alice and Bob generate key pairs where Alice's modulus is one byte larger than Bob's. Then it will frequently be the case that the values in a ciphertext under Alice's public key will be too large to be a ciphertext for Bob, revealing Alice as the recipient.

Key anonymity also needs to be defined and proven for the universal re-encryption operation. We show this by way of example. Let Alice and Bob be two users that use the same group. Suppose that the universal re-encryption operation uses the encryption of unity to re-encrypt the encryption of the message. But, suppose that the re-encryption operation does *not* use the encryption of unity to re-encrypt itself. It follows that the re-encryption operation would never change the encryption of unity. The distinguishing adversary is permitted to choose the ciphertext for the re-encryption operation. The challenger randomly selects Alice or Bob as the receiver and re-encrypts the ciphertext. The adversary is then able to trivially associate the output ciphertext to the receiver by observing the encryption of unity.

Note that both algorithms easily achieve message indistinguishability. However, both the encryption algorithm and the re-encryption algorithm fail to achieve anonymity for different reasons.

4.3 Cryptanalysis of UCS security definitions

Below is the verbatim definition of Golle et al of *universal semantic security under re-encryption*. UKG corresponds to UGEN, UE corresponds to UENCR, and URe corresponds to URENC.

Experiment $\mathbf{EXP}_{\mathcal{A}}^{uss}(UCS, k)$:
1. $PK_0 \leftarrow$ UKG;$PK_1 \leftarrow$ UKG;
2. $(m_0, m_1, r_0, r_1) \leftarrow \mathcal{A}(PK_0, PK_1,$"specify ciphertexts")
3. if $m_0, m_1 \notin \mathbf{M}$ or $r_0, r_1 \notin \mathbf{R}$ then output '0';
4. $C_0 \leftarrow$ UE(m_0, r_0, PK_0);$C_1 \leftarrow$ UE(m_1, r_1, PK_1);
5. $r_0', r_1' \in_U \mathbf{R}$
6. $C_0' \leftarrow$ URe(C_0, r_0');$C_1' \leftarrow$ URe(C_1, r_1');
7. $b \in_U \{0, 1\}$;
8. $b' \leftarrow \mathcal{A}(C_b', C_{1-b}',$"guess");
9. if $b = b'$ then output '1' else output '0';

Why this definition is insufficient: Observe that between the two invocations of the adversary, UE is called immediately followed by URe. The definition therefore does not account for the situation in which an adversary chooses the message to be encrypted with UE under either Alice or Bob's public key and gets to inspect the ciphertext output by UE. Consequently, an algorithm that satisfies this definition will provide no assurance that key anonymity holds for the ciphertexts output by UE. The definition does not reflect the typical use-case of the adversary being able to inspect the ciphertexts output by UE. Furthermore, this is the only definition of security spelled out for the universal re-encryption cryptosystem in [9].

Message indistinguishability of encryption and re-encryption: Golle et al do not define semantic security for the encryption operation nor for the re-encryption operation. Section 3 reads "the properties of standard semantic security and also universal semantic security under re-encryption (as characterized by experiment *uss*) may be shown straightforwardly to be reducible to the Decision Diffie-Hellman (DDH) assumption over the group $\mathcal{G}$, in much the same way as the semantic security of ElGamal". Section 5 indicates that UCS "inherits the semantic security property of the underlying ElGamal cipher under the DDH assumption".

4.4 Cryptanalysis of UCS proofs

No proofs were given for UCS. It was claimed that standard semantic security and also universal semantic security under re-encryption may be shown straightforwardly to be reducible to the Decision Diffie-Hellman (DDH) assumption. The scope of this is incomplete and the claim is not accurate. The scope is incomplete since it does not address key anonymity for the encryption operation. Regarding the "straightforwardly" argument, we are unaware of any prior DDH reduction technique that can be used to prove that key anonymity holds for the encryption operation. The same applies for the re-encryption operation. We give details on this below.

Consequently, proving that anonymity holds for encryption and re-encryption in UCS has been left open. There are additional definitions and security arguments in Golle et al centered around their mix applications. We analyze these security arguments in Section 8.

In hindsight, we believe that the work of Golle et al was an *extremely* insightful step in the right direction of laying

the foundation for universal re-encryption. In particular, we commend their approach of having the adversary fully specify the ciphertexts (messages and nonces) that are used in forming the re-encryption challenge ciphertexts.

4.5 Post-mortem: the non-triviality of proving key anonymity

In light of this analysis it is natural to ask how non-trivial it is to prove that key anonymity holds for a universal cryptosystem. We address this in this subsection. Before we start we need to define precisely what we mean by key anonymity for UENCR. We remark that we use stateful adversaries $\mathcal{A}$ in our definitions of security.

Definition 1. If $\forall$ probabilistic polytime adversaries $\mathcal{A}$, $\forall\ \alpha > 0$, and $\forall$ sufficiently large κ, after the following,

1. generate $((p,q),g) \leftarrow \mathcal{IG}(\kappa)$
2. generate $(y_j, x_j) \leftarrow \mathtt{UGEN}((p,q),g)$ for $j = 0,1$
3. $m \leftarrow \mathcal{A}((p,q),g,y_0,y_1,$"specify message")
4. if $m \notin G_{\mathfrak{p}}$ then output "false" and halt
5. generate $r \in_U [1,q] \times [1,q]$ and $u \in_U \{0,1\}$
6. $c \leftarrow \mathtt{UENCR}(m,r,(y_u,g,p))$
7. $u' \leftarrow \mathcal{A}(c,$"guess")
8. if $u = u'$ then output "true" else output "false"

the output of the experiment is "true" with probability less than $\frac{1}{2} + \frac{1}{\kappa^\alpha}$ then UENCR is key-anonymous.

The adversary is said to succeed if his $u = u'$. Consider the following fatally flawed proof that key anonymity holds for UENCR.

Proof. Suppose there exists a probabilistic polynomial time adversary $\mathcal{A}$, an $\alpha > 0$, and a sufficiently large κ, such that $\mathcal{A}$ succeeds with probability greater than or equal to $\frac{1}{2} + \frac{1}{\kappa^\alpha}$. Consider algorithm AlgA that takes as input a DDH problem instance $((p,q),g,a_0,b_0,c_0)$.

AlgA$((p,q),g,a_0,b_0,c_0)$:
1. set $(\alpha_j, y_j, \mu_j) \leftarrow \mathtt{DDHRerand}((p,q),g,a_0,b_0,c_0)$ for $j = 0,1$
2. $m \leftarrow \mathcal{A}((p,q),g,y_0,y_1,$"specify message")
3. if $m \notin G_{\mathfrak{p}}$ then output "false" and halt
4. generate $u \in_U \{0,1\}$ and $r \in_U [1,q]$
5. set $c \leftarrow ((A_0,B_0),(A_1,B_1)) \leftarrow ((g^r, y_u^r),(\alpha_u, \mu_u m))$
6. $u' \leftarrow \mathcal{A}(c,$"guess")
7. if $u = u'$ then output "true" else output "false"

Consider the case that the input is a DH 3-tuple. It follows from the definition of DDHRerand that c is an encryption of m using y_u as the public key in accordance with UENCR. Therefore, the input to $\mathcal{A}$ is drawn from the same set and probability distribution as the input to $\mathcal{A}$ in Definition 1. It follows that $u = u'$ with probability greater than or equal to $\frac{1}{2} + \frac{1}{\kappa^\alpha}$. So, for random exponents a and b in $[1,q]$, $\Pr[\mathtt{AlgA}((p,q),g,g^a,g^b,g^{ab}) = \text{"true"}] \geq \frac{1}{2} + \frac{1}{\kappa^\alpha}$. Define $\psi = \Pr[\mathtt{AlgA}((p,q),g,g^a,g^b,g^{ab}) = \text{"true"}]$.

Now consider the case that the input is not a DH 3-tuple. It follows from the definition of DDHRerand that the 3-tuple (α_u, y_u, μ_u) is uniformly distributed in $G_{\mathfrak{p}}^3$. Therefore, (A_1,B_1) is uniformly distributed in $G_{\mathfrak{p}}^2$. Since r is randomly chosen, (A_0,B_0) is a proper ElGamal encryption of unity under y_u. Let p_1 be the probability that $\mathcal{A}$ responds with $u' = 0$. Then the probability that $u = u'$ is $\frac{1}{2}p_1 + \frac{1}{2}(1-p_1) = \frac{1}{2}$. So, for randomly chosen exponents a, b, and c in $[1,q]$, $\Pr[\mathtt{AlgA}((p,q),g,g^a,g^b,g^c) = \text{"true"}] = \frac{q^2}{q^3}\psi + (1 - \frac{q^2}{q^3})\frac{1}{2}$ which is overwhelmingly close to $\frac{1}{2}$. □

Observe that if the input is a DH 3-tuple then the reduction algorithm passes a proper universal encryption to the adversary. But if the input is not a DH 3-tuple then an ElGamal encryption of unity and an unconditionally secure encryption of m is passed to the adversary. This 4-tuple is in fact an unconditionally secure encryption of m. The adversary already "knows" that (A_0,B_0) encrypts unity. The problem is that this unconditionally secure encryption of m does not appear to be sufficient to prove that key anonymity holds. The ElGamal encryption of unity could potentially reveal the underlying public key to the adversary. More concretely, the following probabilistic polynomial time adversary could exist:

1. $\mathcal{A}$ computes the base g discrete logarithm of A_0 to get r
2. $\mathcal{A}$ computes $y_u = B_0^{r^{-1}} \bmod p$
3. $\mathcal{A}$ outputs u "with non-negligible advantage"

The argument therefore cannot be made that "$\mathcal{A}$ can do no better than guess u". With this adversary there is no polytime observable difference in behavior with which to solve DDH.

Alternatively, suppose that the reduction algorithm invokes DDHRerand twice on the input problem instance in an effort to unconditionally hide the whole universal ciphertext. The problem here is that the common "public key" y no longer exists[1] between the two output 3-tuples.

The same challenge arises in proving that key anonymity holds for the re-encryption operation. So, we believe that this demonstrates that proving that key anonymity holds for encryption and re-encryption is non-trivial. We solve this problem by generalizing the DDH random self-reduction.

5. NEW CONSTRUCTION: EXPANDED DDH SELF-REDUCTION

We now generalize the DDH random self-reduction to output five values instead of three. This allows us to transform a DDH problem instance into either two DH 3-tuples with a common "public key" or a random 5-tuple, depending on the input problem instance. We utilize this feature in our proofs of security in Sections 6 and 7. We define algorithm DDHRerand5 as follows. DDHRerand5$((p,q),g,x,y,z)$ randomizes a DDH problem instance by choosing the values $u_1,u_2,v,v',u_1' \in_U [1,q]$ and computing,

$$(x'',x',y',z',z'') \leftarrow$$
$$(x^{v'}g^{u_1'}, x^v g^{u_1}, y g^{u_2}, z^v y^{u_1} x^{vu_2} g^{u_1 u_2}, z^{v'} y^{u_1'} x^{v'u_2} g^{u_1' u_2})$$

Case 1. Suppose (x,y,z) is a valid Diffie-Hellman (DH) 3-tuple. Then $x = g^a$, $y = g^b$, $z = g^{ab}$ for some a,b. It follows that (x',y',z') is also a valid DH 3-tuple. It is straightforward to show that (x'',y',z'') is a valid DH 3-tuple as well.

[1]With overwhelming probability.

Case 2. Suppose (x, y, z) is not a valid DH 3-tuple. Then $x = g^a$, $y = g^b$, $z = g^{ab+c}$ for some $c \neq 0$. In this case, $x' = g^{a'}$, $y' = g^{b'}$, $z' = g^{a'b'}g^{cv}$. Since $c \neq 0$ it follows that g^c is a generator of $G_{\mathfrak{p}}$. Also, $x'' = g^{a''}$, $y' = g^{b'}$, $z'' = g^{a''b'}g^{cv'}$.

So, when (x, y, z) is a valid DH 3-tuple then (x', y', z') and (x'', y', z'') are random DH 3-tuples with y' in common and when (x, y, z) is not a valid DH 3-tuple then the output is a random 5-tuple.

6. PROVABLY SECURE UNIVERSAL RE-ENCRYPTION

6.1 Our re-encryption algorithm

We now present our modification to UCS that we call URE. It is equivalent to UCS with the exception of how the re-randomization operation is performed.

Universal Re-encryption: The following is the universal re-encryption operation,

$$\texttt{URENCR}(((a_0, b_0), (a_1, b_1)), (\ell_0, \ell_1), p)$$

The pair $((a_0, b_0), (a_1, b_1))$ is a universal ciphertext produced using UENCR and the value $(\ell_0, \ell_1) \in_U [1, q] \times [1, q]$ is a pair of re-encryption nonces. Compute $(\alpha_0, \beta_0) \leftarrow (a_0 a_0^{\ell_0} \bmod p, b_0 b_0^{\ell_0} \bmod p)$ and compute $(\alpha_1, \beta_1) \leftarrow (a_1 a_0^{\ell_1} \bmod p, b_1 b_0^{\ell_1} \bmod p)$. Output the ciphertext $c \leftarrow ((\alpha_0, \beta_0), (\alpha_1, \beta_1))$.

The encryption nonces are parameterized to facilitate our proofs of security. The URE cryptosystem is the 4-tuple (UGEN, UENCR, URENCR, UDECR).

Note the difference between URENC and URENCR. In URENCR we perform an additional pairwise multiplication by (a_0, b_0). We do so since we believe it makes our reduction proofs more straightforward than had we used URENC.

6.2 Security proof of universal re-encryption

In this section we prove that URE achieves semantically secure anonymity that we define as follows.

Definition 2. If a universal public key cryptosystem URE consisting of algorithms UGEN, UENCR, URENCR, and UDECR has the following properties:

1. UENCR satisfies message indistinguishability.
2. UENCR satisfies key anonymity.
3. URENCR satisfies message indistinguishability.
4. URENCR satisfies key anonymity.

then URE is secure in the sense of ***semantically secure anonymity***.

The below message indistinguishability definition has been adapted from [7, 16].

Definition 3. If $\forall$ probabilistic polytime adversaries $\mathcal{A}$, $\forall\ \alpha > 0$, and $\forall$ sufficiently large κ, after the following,

1. generate $((p, q), g) \leftarrow \mathcal{IG}(\kappa)$
2. compute $(y, x) \leftarrow \texttt{UGEN}((p, q), g)$
3. $(m_0, m_1) \leftarrow \mathcal{A}((p, q), g, y,$"specify messages")
4. if ($m_0 \notin G_{\mathfrak{p}}$ or $m_1 \notin G_{\mathfrak{p}}$ or $m_0 = m_1$) then output "false" and halt
5. $(k_0, k_1) \in_U [1, q] \times [1, q]$
6. $b \in_U \{0, 1\}$, compute $c \leftarrow \texttt{UENCR}(m_b, (k_0, k_1), (y, g, p))$
7. $b' \leftarrow \mathcal{A}(c,$"guess")
8. if $b = b'$ then output "true" else output "false"

the output of the experiment is "true" with probability less than $\frac{1}{2} + \frac{1}{\kappa^\alpha}$ then UENCR is secure in the sense of message indistinguishability.

THEOREM 1. *If DDH is hard then* UENCR *is secure in the sense of message indistinguishability.*

PROOF. Suppose there exists a probabilistic polynomial time adversary $\mathcal{A}$, an $\alpha > 0$, and a sufficiently large κ, such that $\mathcal{A}$ succeeds with probability greater than or equal to $\frac{1}{2} + \frac{1}{\kappa^\alpha}$. Consider algorithm AlgR1 that takes as input a DDH problem instance $((p, q), g, a_0, b_0, c_0)$.

AlgR1$((p, q), g, a_0, b_0, c_0)$:
1. set $(A', A, y, R, R') \leftarrow$ DDHRerand5$((p, q), g, a_0, b_0, c_0)$
2. $(m_0, m_1) \leftarrow \mathcal{A}((p, q), g, y,$"specify messages")
3. if ($m_0 \notin G_{\mathfrak{p}}$ or $m_1 \notin G_{\mathfrak{p}}$ or $m_0 = m_1$) then output "false" and halt
4. $b \in_U \{0, 1\}$
5. $c \leftarrow ((A_0, B_0), (A_1, B_1)) \leftarrow ((A', R'), (A, Rm_b))$
6. $b' \leftarrow \mathcal{A}(c,$"guess")
7. if $b = b'$ then output "true" else output "false"

Consider the case that the input is a DH 3-tuple. It follows from the definition of DDHRerand5 that c is an encryption of m_b according to UENCR using y as the public key. Therefore, the input to $\mathcal{A}$ is drawn from the same set and probability distribution as the input to $\mathcal{A}$ in Definition 3. It follows that $b = b'$ with probability greater than or equal to $\frac{1}{2} + \frac{1}{\kappa^\alpha}$. So, for random exponents a and b in $[1, q]$, $\Pr[\texttt{AlgR1}((p, q), g, g^a, g^b, g^{ab}) =$ "true"$] \geq \frac{1}{2} + \frac{1}{\kappa^\alpha}$. Define $\psi = \Pr[\texttt{AlgR1}((p, q), g, g^a, g^b, g^{ab}) =$ "true"$]$.

Now consider the case that the input is not a DH 3-tuple. It follows from the definition of DDHRerand5 that (A', A, y, R, R') is uniformly distributed in $G_{\mathfrak{p}}^5$. Therefore, c is uniformly distributed in $G_{\mathfrak{p}}^2 \times G_{\mathfrak{p}}^2$. Let p_1 be the probability that $\mathcal{A}$ responds with $b' = 0$. Then the probability that $b = b'$ is $\frac{1}{2}p_1 + \frac{1}{2}(1 - p_1) = \frac{1}{2}$. So, for randomly chosen exponents a, b, and c in $[1, q]$, the probability $\Pr[\texttt{AlgR1}((p, q), g, g^a, g^b, g^c) =$ "true"$] = \frac{q^2}{q^3}\psi + (1 - \frac{q^2}{q^3})\frac{1}{2}$ which is overwhelmingly close to $\frac{1}{2}$. □

We now prove that URENCR is secure in the sense of message indistinguishability.

Definition 4. If $\forall$ probabilistic poly-time adversaries $\mathcal{A}$, $\forall\ \alpha > 0$, and $\forall$ sufficiently large κ, after the following,

1. generate $((p, q), g) \leftarrow \mathcal{IG}(\kappa)$, $(y, x) \leftarrow \texttt{UGEN}((p, q), g)$
2. $(m_0, m_1, r_0, r_1) \leftarrow \mathcal{A}((p, q), g, y,$"specify ciphertexts")
3. if ($(m_0, m_1) \notin G_{\mathfrak{p}} \times G_{\mathfrak{p}}$ or $m_0 = m_1$) then output "false" and halt
4. if ($r_0 \notin [1, q] \times [1, q]$ or $r_1 \notin [1, q] \times [1, q]$) then output "false" and halt
5. $b \in_U \{0, 1\}$
6. $c \leftarrow \texttt{UENCR}(m_b, r_b, (y, g, p))$
7. $r \in_U [1, q] \times [1, q]$
8. $c' \leftarrow \texttt{URENCR}(c, r, p)$
9. $b' \leftarrow \mathcal{A}(c',$"guess")
10. if $b = b'$ then output "true" else output "false"

the output of the experiment is "true" with probability less than $\frac{1}{2} + \frac{1}{\kappa^\alpha}$ then URENCR is secure in the sense of message indistinguishability.

THEOREM 2. *If DDH is hard then* URENCR *is secure in the sense of message indistinguishability.*

PROOF. Suppose there exists a probabilistic polynomial time adversary $\mathcal{A}$, an $\alpha > 0$, and a sufficiently large κ, such that $\mathcal{A}$ succeeds with probability greater than or equal to $\frac{1}{2} + \frac{1}{\kappa^\alpha}$. Consider algorithm AlgR2 that takes as input a DDH problem instance $((p,q), g, a_0, b_0, c_0)$.

AlgR2$((p,q), g, a_0, b_0, c_0)$:
1. set $(\alpha', \alpha, y, \mu, \mu') \leftarrow$ DDHRerand5$((p,q), g, a_0, b_0, c_0)$
2. $(m_0, m_1, r_0, r_1) \leftarrow \mathcal{A}((p,q), g, y,$"specify ciphertexts")
3. if $((m_0, m_1) \notin G_\mathfrak{p} \times G_\mathfrak{p}$ or $m_0 = m_1)$ then output "false" and halt
4. if $(r_0 \notin [1,q] \times [1,q]$ or $r_1 \notin [1,q] \times [1,q])$ then output "false" and halt
5. $b \in_U \{0,1\}$
6. $((A_0, B_0), (A_1, B_1)) \leftarrow$ UENCR$(m_b, r_b, (y, g, p))$
7. $c' \leftarrow ((A_0\alpha', B_0\mu'), (A_1\alpha, B_1\mu))$
8. $b' \leftarrow \mathcal{A}(c',$"guess")
9. if $b = b'$ then output "true" else output "false"

Consider the case that the input is a DH 3-tuple. Clearly $((A_0, B_0), (A_1, B_1))$ is the ciphertext of m_b as specified by adversary $\mathcal{A}$. It follows from the definition of DDHRerand5 that c' is a re-encryption of $((A_0, B_0), (A_1, B_1))$ according to URENCR. Therefore, the input to $\mathcal{A}$ is drawn from the same set and probability distribution as the input to $\mathcal{A}$ in Definition 4. It follows that $b = b'$ with probability greater than or equal to $\frac{1}{2} + \frac{1}{\kappa^\alpha}$. So, for random exponents a and b in $[1,q]$, $\Pr[\text{AlgR2}((p,q), g, g^a, g^b, g^{ab}) = \text{"true"}] \geq \frac{1}{2} + \frac{1}{\kappa^\alpha}$. Define the value ψ to be $\Pr[\text{AlgR2}((p,q), g, g^a, g^b, g^{ab}) = \text{"true"}]$.

Now consider the case that the input is not a DH 3-tuple. It follows from the definition of DDHRerand5 that $(\alpha', \alpha, y, \mu, \mu')$ is uniformly distributed in the set $G_\mathfrak{p}^5$. Therefore, c' is uniformly distributed in $G_\mathfrak{p}^2 \times G_\mathfrak{p}^2$. Let p_1 be the probability that $\mathcal{A}$ responds with $b' = 0$. Then the probability that $b = b'$ is $\frac{1}{2}p_1 + \frac{1}{2}(1 - p_1) = \frac{1}{2}$. So, for randomly chosen exponents a, b, and c in $[1,q]$, the probability $\Pr[\text{AlgR2}((p,q), g, g^a, g^b, g^c) = \text{"true"}] = \frac{q^2}{q^3}\psi + (1 - \frac{q^2}{q^3})\frac{1}{2}$ which is overwhelmingly close to $\frac{1}{2}$. □

We now prove that UENCR is key-anonymous.

THEOREM 3. *If DDH is hard then algorithm* UENCR *is key-anonymous.*

PROOF. Suppose there exists a probabilistic polynomial time adversary $\mathcal{A}$, an $\alpha > 0$, and a sufficiently large κ, such that $\mathcal{A}$ succeeds with probability greater than or equal to $\frac{1}{2} + \frac{1}{\kappa^\alpha}$. Consider algorithm AlgR3 that takes as input a DDH problem instance $((p,q), g, a_0, b_0, c_0)$.

AlgR3$((p,q), g, a_0, b_0, c_0)$:
1. set $(\alpha'_j, \alpha_j, y_j, \mu_j, \mu'_j) \leftarrow$ DDHRerand5$((p,q), g, a_0, b_0, c_0)$ for $j = 0, 1$
2. $m \leftarrow \mathcal{A}((p,q), g, y_0, y_1,$"specify message")
3. if $m \notin G_\mathfrak{p}$ then output "false" and halt
4. generate $u \in_U \{0,1\}$
5. set $c \leftarrow ((A_0, B_0), (A_1, B_1)) \leftarrow ((\alpha'_u, \mu'_u), (\alpha_u, \mu_u m))$
6. $u' \leftarrow \mathcal{A}(c,$"guess")
7. if $u = u'$ then output "true" else output "false"

Consider the case that the input is a DH 3-tuple. It follows from the definition of DDHRerand5 that c is an encryption of m in accordance with UENCR using y_u as the public key. Therefore, the input to $\mathcal{A}$ is drawn from the same set and probability distribution as the input to $\mathcal{A}$ in Definition 1. It follows that $u = u'$ with probability greater than or equal to $\frac{1}{2} + \frac{1}{\kappa^\alpha}$. So, for random exponents a and b in $[1,q]$, $\Pr[\text{AlgR3}((p,q), g, g^a, g^b, g^{ab}) = \text{"true"}] \geq \frac{1}{2} + \frac{1}{\kappa^\alpha}$. Define ψ = $\Pr[\text{AlgR3}((p,q), g, g^a, g^b, g^{ab}) = \text{"true"}]$.

Now consider the case that the input is not a DH 3-tuple. It follows from the definition of DDHRerand5 that the 5-tuple $(\alpha'_u, \alpha_u, y_u, \mu_u, \mu'_u)$ is uniformly distributed in $G_\mathfrak{p}^5$. Therefore, c is uniformly distributed in $G_\mathfrak{p}^2 \times G_\mathfrak{p}^2$. Let p_1 be the probability that $\mathcal{A}$ responds with $u' = 0$. Then the probability that $u = u'$ is $\frac{1}{2}p_1 + \frac{1}{2}(1 - p_1) = \frac{1}{2}$. So, for randomly chosen exponents a, b, and c in $[1,q]$, the probability $\Pr[\text{AlgR3}((p,q), g, g^a, g^b, g^c) = \text{"true"}] = \frac{q^2}{q^3}\psi + (1 - \frac{q^2}{q^3})\frac{1}{2}$ which is overwhelmingly close to $\frac{1}{2}$. □

We now prove that URENCR is key-anonymous.

Definition 5. If $\forall$ probabilistic polytime adversaries $\mathcal{A}$, $\forall\ \alpha > 0$, and $\forall$ sufficiently large κ, after the following,

1. generate $((p,q), g) \leftarrow \mathcal{IG}(\kappa)$
2. $(y_j, x_j) \leftarrow$ UGEN$((p,q), g)$ for $j = 0, 1$
3. $(m, (k_0, k_1)) \leftarrow \mathcal{A}((p,q), g, y_0, y_1,$"specify ciphertext")
4. if $(m \notin G_\mathfrak{p}$ or $(k_0, k_1) \notin [1,q] \times [1,q])$ then output "false" and halt
5. $u \in_U \{0,1\}$
6. $c \leftarrow ((a_0, b_0), (a_1, b_1)) \leftarrow$ UENCR$(m, (k_0, k_1), (y_u, g, p))$
7. generate $r \in_U [1,q] \times [1,q]$
8. set $c' \leftarrow$ URENCR(c, r, p)
9. $u' \leftarrow \mathcal{A}(c',$"guess")
10. if $u = u'$ then output "true" else output "false"

the output of the experiment is "true" with probability less than $\frac{1}{2} + \frac{1}{\kappa^\alpha}$ then URENCR is key-anonymous.

THEOREM 4. *If DDH is hard then the algorithm* URENCR *is key-anonymous.*

PROOF. Suppose there exists a probabilistic polynomial time adversary $\mathcal{A}$, an $\alpha > 0$, and a sufficiently large κ such that $\mathcal{A}$ succeeds with probability greater than or equal to $\frac{1}{2} + \frac{1}{\kappa^\alpha}$. Consider algorithm AlgR4 that takes as input a Decision Diffie-Hellman problem instance $((p,q), g, a_0, b_0, c_0)$.

AlgR4$((p,q), g, a_0, b_0, c_0)$:
1. $(\alpha'_j, \alpha_j, y_j, \mu_j, \mu'_j) \leftarrow$ DDHRerand5$((p,q), g, a_0, b_0, c_0)$ for $j = 0, 1$
2. $(m, (k_0, k_1)) \leftarrow \mathcal{A}((p,q), g, y_0, y_1,$"specify ciphertext")
3. if $(m \notin G_\mathfrak{p}$ or $(k_0, k_1) \notin [1,q] \times [1,q])$ then output "false" and halt
4. $u \in_U \{0,1\}$
5. $((A_0, B_0), (A_1, B_1)) \leftarrow$ UENCR$(m, (k_0, k_1), (y_u, g, p))$
6. $c' \leftarrow ((A'_0, B'_0), (A'_1, B'_1)) \leftarrow ((A_0\alpha'_u, B_0\mu'_u), (A_1\alpha_u, B_1\mu_u))$
7. $u' \leftarrow \mathcal{A}(c',$"guess")
8. if $u = u'$ then output "true" else output "false"

Consider the case that the input is a DH 3-tuple. Clearly $((A_0, B_0), (A_1, B_1))$ is the ciphertext under public key y_u as specified by $\mathcal{A}$. It follows from the definition of DDHRerand5 that c' is a re-encryption of $((A_0, B_0), (A_1, B_1))$ in accordance with URENCR. Therefore, the input to $\mathcal{A}$ is drawn from the same set and probability distribution as the input to $\mathcal{A}$ in

Definition 5. It follows that $u = u'$ with probability greater than or equal to $\frac{1}{2}+\frac{1}{\kappa^\alpha}$. So, for random exponents a and b in $[1,q]$, $\Pr[\texttt{AlgR4}((p,q),g,g^a,g^b,g^{ab}) = \text{“true”}] \geq \frac{1}{2}+\frac{1}{\kappa^\alpha}$. Define the value ψ to be $\Pr[\texttt{AlgR4}((p,q),g,g^a,g^b,g^{ab}) = \text{“true”}]$.

Now consider the case that the input is not a DH 3-tuple. It follows from definition of DDHRerand5 that the 5-tuple $(\alpha'_u, \alpha_u, y_u, \mu_u, \mu'_u)$ is uniformly distributed in $G_{\mathfrak{p}}^5$. Therefore, c' is uniformly distributed in $G_{\mathfrak{p}}^2 \times G_{\mathfrak{p}}^2$. Let p_1 be the probability that $\mathcal{A}$ responds with $u' = 0$. Then the probability that $u = u'$ is $\frac{1}{2}p_1 + \frac{1}{2}(1-p_1) = \frac{1}{2}$. So, for randomly chosen exponents a, b, and c in $[1,q]$, the probability $\Pr[\texttt{AlgR4}((p,q),g,g^a,g^b,g^c) = \text{“true”}] = \frac{q^2}{q^3}\psi + (1-\frac{q^2}{q^3})\frac{1}{2}$ which is overwhelmingly close to $\frac{1}{2}$. □

Theorems 1, 2, 3, and 4 show the following.

THEOREM 5. *If DDH is hard then* URE *is secure in the sense of semantically secure anonymity.*

7. CRYPTANALYSIS OF INCOMPARABLE PUBLIC KEYS

An incomparable public key has the property that it can be changed over time to conceal the owner of the key. This can be used to hide the identity of receivers from message senders. In this section we review the security foundation of the ElGamal-based incomparable public key cryptosystem, cryptanalyze it, and then present the first proof that it is secure under DDH.

Define $\texttt{IGEN}((p,q),g)$ to be an algorithm that outputs the tuple $((g^k, y^k), x)$ where $y = g^x \bmod p$ and $k, x \in_U [1,q]$. Let (a,b) denote the incomparable public key (g^k, y^k). The corresponding private key is x. Let $\texttt{IRR}((p,q),(a,b))$ denote the incomparable public key re-randomization function. This function outputs $(a^r \bmod p, b^r \bmod p)$ where $r \in_U [1,q]$.

Waters et al claim in the introduction of [21] that they prove that the cryptographic properties of their cryptosystem hold under the assumption that DDH is hard. The appendices present the definition and proofs of incomparability and key-privacy. The reduction proofs define experiments that take as input algorithms and a “common key” (which in the case of ElGamal is the global ElGamal modulus p). Nowhere in their reduction algorithms do they take a DDH problem instance as input. Consequently, the security of the incomparable public key cryptosystem was not tied to DDH in any meaningful way. Therefore, security was not shown to hold under DDH. We now prove that the incomparability property holds under DDH.

Definition 6. If $\forall$ probabilistic polytime adversaries $\mathcal{A}$, $\forall\ \alpha > 0$, and $\forall$ sufficiently large κ, after the following,

1. generate $((p,q),g) \leftarrow \mathcal{IG}(\kappa)$
2. $((a_i,b_i),x_i) \leftarrow \texttt{IGEN}((p,q),g)$ for $i = 0,1$
3. $t \in_U \{0,1\}, y_0 = g^{x_0} \bmod p, y_1 = g^{x_1} \bmod p$
4. $(a_2,b_2) \leftarrow \texttt{IRR}((p,q),(a_t,b_t))$
5. $t' \leftarrow \mathcal{A}((p,q),g,(a_0,b_0,y_0),(a_1,b_1,y_1),(a_2,b_2))$
6. if $t = t'$ then output “true” else output “false”

the output of the experiment is “true” with probability less than $\frac{1}{2}+\frac{1}{\kappa^\alpha}$ then $(\texttt{IGEN},\texttt{IRR})$ is secure in the sense of incomparability.

Note that we give the adversary y_0 and y_1. This is information that adversaries in practice might not have. But it is perfectly okay to give it to the adversary as auxiliary information.

THEOREM 6. *If DDH is hard then* $(\texttt{IGEN},\texttt{IRR})$ *is secure in the sense of incomparability.*

PROOF. Suppose there exists a probabilistic polynomial time adversary $\mathcal{A}$, an $\alpha > 0$, and a sufficiently large κ, such that $\mathcal{A}$ succeeds with probability greater than or equal to $\frac{1}{2}+\frac{1}{\kappa^\alpha}$. Consider algorithm AlgB that takes as input a DDH problem instance $((p,q),g,a_0,b_0,c_0)$.

$\texttt{AlgB}((p,q),g,a_0,b_0,c_0)$:
1. $(A'_0,A_0,y_0,R_0,R'_0) \leftarrow \texttt{DDHRerand5}((p,q),g,a_0,b_0,c_0)$
2. $(A'_1,A_1,y_1,R_1,R'_1) \leftarrow \texttt{DDHRerand5}((p,q),g,a_0,b_0,c_0)$
3. $t \in_U \{0,1\}$
4. $t' \leftarrow \mathcal{A}((p,q),g,(A_0,R_0,y_0),(A_1,R_1,y_1),(A'_t,R'_t))$
5. if $t = t'$ then output “true” else output “false”

Consider the case that the input is a DH 3-tuple. It follows from the definition of DDHRerand5 that (A_i,R_i) is a proper incomparable public key with underlying trapdoor value y_i under IGEN for $i = 0,1$. It also follows that (A'_t,R'_t) is a proper re-randomization of (A_t,R_t) under IRR. Therefore, the input to $\mathcal{A}$ is drawn from the same set and probability distribution as the input to $\mathcal{A}$ in Definition 6. It follows that $t = t'$ with probability greater than or equal to $\frac{1}{2}+\frac{1}{\kappa^\alpha}$. So, for random exponents a and b in $[1,q]$, $\Pr[\texttt{AlgB}((p,q),g,g^a,g^b,g^{ab}) = \text{“true”}] \geq \frac{1}{2}+\frac{1}{\kappa^\alpha}$. Define $\psi = \Pr[\texttt{AlgB}((p,q),g,g^a,g^b,g^{ab}) = \text{“true”}]$.

Now consider the case that the input is not a DH 3-tuple. It follows from the definition of DDHRerand5 that $(A_0,R_0,y_0,A_1,R_1,y_1,A'_t,R'_t)$ is uniformly distributed in $G_{\mathfrak{p}}^8$. Let p_1 be the probability that $\mathcal{A}$ responds with $t' = 0$. Then the probability that $t = t'$ is $\frac{1}{2}p_1 + \frac{1}{2}(1-p_1) = \frac{1}{2}$. So, for randomly chosen exponents a, b, and c in $[1,q]$, the probability $\Pr[\texttt{AlgB}((p,q),g,g^a,g^b,g^c) = \text{“true”}] = \frac{q^2}{q^3}\psi + (1-\frac{q^2}{q^3})\frac{1}{2}$ which is overwhelmingly close to $\frac{1}{2}$. □

Collectively, the security arguments for UCS and incomparable cryptosystems point towards a trend of claiming security with respect to DDH but not proving it. We hope that this observation and the reduction techniques we presented will help improve future work.

We point out that both UCS and incomparable public keys exhibit the same challenge in proving that they achieve the anonymity property. In the case of UCS the reduction algorithm presents to the adversary an encryption of unity and an encryption of a message under a common public key. For the incomparable public key proof the reduction algorithm presents to the adversary two encryptions of unity under a common public key. DDHRerand5 allows the 4-tuple to be: two ElGamal encryptions under a common key, or a random 4-tuple depending on the DDH problem instance. It does not appear that the classic DDH self-reduction allows for this.

8. FORWARD-ANONYMOUS BATCH MIX

Golle et al used the universal cryptosystem to construct a forward anonymous mix protocol centered around the use

of a bulletin board. The number of ciphertexts on the board can vary over time. Servers download the ciphertexts from the board, re-randomize them, and then upload them in permuted order. We instead chose to analyze a batch mix that mixes a fixed number of ciphertexts. We consider this case since: (1) it is concrete in the sense that a fixed size vector of ciphertexts needs to be anonymized and this gives a precise level of anonymity (fixed-size random permutation), and (2) we achieve low-latency since once the batch forms at the first mix the ciphertexts are pushed through the cascade of mixes rapidly.

We point out that the security arguments of Golle et al for their proposed mixes are flawed:

1. **Not tied to DDH**: None of the proofs in the paper take a DDH problem instance as input. It follows that they did not prove that security holds under DDH.

2. **Not randomized reductions**: None of the input problem instances in the paper are randomized. It is well-known that randomized reductions are stronger than non-randomized ones.

Consequently the security of their mixes were not tied to the DDH problem as claimed. This left as open the problem of proving the security of universal re-encryption batch mixing. We solve this problem in this section.

Informally, the problem we consider is to establish an externally anonymous communication channel. A set of w senders $s_1, s_2, ..., s_w$ want to send messages $m_1, m_2, ..., m_w$ respectively, to a target set of w receivers $r_1, r_2, ..., r_w$. Consider the case that s_i sends a message to s_j where $i, j \in \{1, 2, ..., w\}$. We want an eavesdropper to have negligible advantage in correlating the initial ciphertext that s_i sends out with the public key of r_j. In other words, the eavesdropper has negligible advantage over guessing the receiver.

The solution must be forward-anonymous: an adversary that compromises a mix server cannot break the anonymity of previously transmitted ciphertexts. The solution must be robust in that anonymity holds as long as there is at least one mix server not compromised by the adversary.

Note that a receiver of a message can determine who the sender of the message is. The receiver is able to decipher the ciphertext right when the sender transmits it to the first mix. Anonymity is against external adversaries.

Definition 7. A ***forward-anonymous batch mix*** protocol, denoted by `FBMIX`, is a 4-tuple of algorithms `FBGEN`, `FBENCR`, `FBMIXER`, and `FBDECR` where `FBGEN` generates a key pair for each receiver, where `FBENCR` encrypts the messages of the senders, where the `FBMIXER` servers are connected in series and they mix received ciphertexts and forward them on, that satisfies the following properties for all probabilistic polynomial-time passive adversaries $\mathcal{A}$:

1. `FBENCR` Confidentiality: The ciphertexts output by algorithm `FBENCR` satisfy the message indistinguishability property with respect to $\mathcal{A}$ (Definition 10).

2. `FBMIXER` Confidentiality: The ciphertexts output by `FBMIXER` satisfy message indistinguishability with respect to $\mathcal{A}$ (Definition 11)

3. `FBENCR` Anonymity: The ciphertexts output by `FBENCR` satisfy key anonymity with respect to $\mathcal{A}$ (Definition 8).

4. `FBMIXER` Anonymity: The ciphertexts output by algorithm `FBMIXER` satisfy anonymity with respect to $\mathcal{A}$ (Definition 9).

5. Forward-Anonymity: The `FBMIXER` servers have no secret key material.

6. Robustness: Anonymity of `FBMIX` holds provided at least one `FBMIXER` server is not compromised by $\mathcal{A}$.

7. Completeness: $\forall\, i \in \{1, 2, ..., w\}$, when sender s_i sends m_i to r_j where $j \in \{1, 2, ..., w\}$ then r_j receives m_i.

8. Low-Latency: Once w ciphertexts arrive at the first `FBMIXER` server, the batch moves through the mix at a speed limited only by the time to re-encrypt, permute, and forward.

We instantiate the mix as follows.

`FBGEN`$((p, q), g)$:
1. $(y_i, x_i) \leftarrow$ `UGEN`$((p, q), g)$ for $i = 1, 2, ..., w$
2. output $((y_1, x_1), (y_2, x_2), ..., (y_w, x_w))$

Let σ_i be the index of the receiver of the message of sender i. For example, if s_1 sends to s_3 then $\sigma_1 = 3$.

`FBENCR`$(p, g, (m_1, (k_{1,0}, k_{1,1}), \sigma_1), ...,$
$(m_w, (k_{w,0}, k_{w,1}), \sigma_w), y_1, y_2, ..., y_w)$:
1. $c_i \leftarrow$ `UENCR`$(m_i, (k_{i,0}, k_{i,1}), (y_{\sigma_i}, g, p))$ for $i = 1, 2, ..., w$
2. output $(c_1, c_2, ..., c_w)$

Define set S to be $\{1, 2, ..., w\}$. Let π be a permutation from S onto S. Define `fp`$(\pi, c_1, c_2, ..., c_w)$ to be a function that outputs $(c_{\pi(1)}, c_{\pi(2)}, ..., c_{\pi(w)})$. Let the algorithm `fpinv`$(\pi, c_{\pi(1)}, c_{\pi(2)}, ..., c_{\pi(w)})$ be a function that uses π^{-1} to output the tuple $(c_1, c_2, ..., c_w)$.

`FBMIXER`$(p, \pi, (c_1, (\ell_{1,0}, \ell_{1,1})), (c_2, (\ell_{2,0}, \ell_{2,1})), ...,$
$(c_w, (\ell_{w,0}, \ell_{w,1}))$:
1. $c'_i \leftarrow$ `URE`$(c_i, (\ell_{i,0}, \ell_{i,1}), p)$ for $i = 1, 2, ..., w$
2. output `fp`$(\pi, c'_1, c'_2, ..., c'_w)$

The `break` statement terminates the execution of the nearest enclosing `for` loop in which `break` appears.

`FBDECR`$(p, c_1, c_2, ..., c_w, x_1, x_2, ..., x_w)$:
1. let L be the empty list
2. `for` i in 1 to w:
3. `for` j in 1 to w:
4. $(m, s) \leftarrow$ `UDECR`(c_i, x_j, p)
5. if $s =$ true
6. append (m, j) to L
7. `break`
8. output L

There are four stages in the mix protocol. The mix protocol leverages N mix servers labeled $1, 2, ..., N$ and they are connected in series.

Stage 1: r_j generates a key pair (y_j, x_j) using `UGEN` and publishes y_j for $j = 1, 2, ..., w$. This stage is effectively `FBGEN`.

Stage 2: Sender s_i formulates a message m_i to send to receiver r_j. s_i generates $(k_{i,0}, k_{i,1}) \in_U [1, q] \times [1, q]$ and computes $c_i \leftarrow$ `UENCR`$(m_i, (k_{i,0}, k_{i,1}), (y_{\sigma_i}, g, p))$. s_i sends c_i to Mix 1 for $i = 1, 2, ..., w$. This stage is effectively `FBENCR`.

Stage 3: Mix k where $1 \leq k \leq N$ operates as follows. It waits until a full batch of w ciphertexts $c_1, c_2, ..., c_w$ arrive. It then generates $(\ell_{i,0}, \ell_{i,1}) \in_U [1,q] \times [1,q]$ for $i = 1, 2, ..., w$. It generates a permutation π from S onto S uniformly at random. It then computes,

$$(c'_{\pi(1)}, c'_{\pi(2)}, ..., c'_{\pi(w)}) \leftarrow \texttt{FBMIXER}(p, \pi, (c_1, (\ell_{1,0}, \ell_{1,1})), (c_2, (\ell_{2,0}, \ell_{2,1})), ..., (c_w, (\ell_{w,0}, \ell_{w,1})))$$

If $k < N$ then $(c'_{\pi(1)}, c'_{\pi(2)}, ..., c'_{\pi(w)})$ is sent to mix $k+1$. If $k = N$ then $(c'_{\pi(1)}, c'_{\pi(2)}, ..., c'_{\pi(w)})$ is posted to a public bulletin board. Each of these mixes is effectively FBMIXER.

Stage 4: r_j for $j = 1, 2, ..., w$ downloads all w ciphertexts from the bulletin board. r_j attempts decryption of every single one of the ciphertexts using x_j. In so doing, r_j receives zero or more messages. If there is no i for which $\sigma_i = j$ then r_j receives no messages. This stage is effectively FBDECR.

We can improve the performance of Stage 4 in the case that every receiver gets only one message from a sender. In this scenario, a receiver can pull down the ciphertexts from the bulletin board one by one and then stop when a ciphertext is received that properly decrypts. The batch mix provides external anonymity thereby breaking the link between senders and receivers. This use case would fail completely were the senders to post their key anonymous ciphertexts directly to the bulletin board. To see this, note that a passive eavesdropper would know the sender of each ciphertext on the bulletin board. The eavesdropper would then know who the receiver is of a given ciphertext based on when the receiver stops pulling down ciphertexts.

9. SECURITY OF FBMIX

Where possible we allow the adversary to choose the receivers of messages in FBMIX. For example, the adversary can have Alice and Bob send messages to the same receiver, Carol. Consequently, many senders can send messages to the same receiver. As a result we need to generalize DDHRerand5 from Section 5. It generalizes to produce more DH 3-tuples with a common "public key" in the same way that the DDH random self-reduction generalized to form DDHRerand5.

To make the pattern clear we define DDHRerand7 as follows. The algorithm $\texttt{DDHRerand7}((p,q), g, x, y, z)$ randomizes a DDH problem instance by choosing the exponents $u_1, u_2, v, v', v'', u'_1, u''_1 \in_U [1,q]$ and computing,

$$(x''', x'', x', y', z', z'', z''') \leftarrow (x^{v''} g^{u''_1}, x^{v'} g^{u'_1}, x^v g^{u_1}, y g^{u_2}, z^v y^{u_1} x^{v u_2} g^{u_1 u_2}, z^{v'} y^{u'_1} x^{v' u_2} g^{u'_1 u_2}, z^{v''} y^{u''_1} x^{v'' u_2} g^{u''_1 u_2})$$

and so on for ever more "v primes" and "u_1 primes".

For ease of use we parameterize this DDH generalization as follows. Let $\texttt{DDHRerandN}((p,q), g, x, y, z, t)$ be a DDH self-reduction algorithm that outputs a set T containing t 3-tuples. Define, the set $T = \{(A_1, B_1, R_1), (A_2, B_2, R_2), ..., (A_t, B_t, R_t)\}$.

The algorithm has these properties: (1) when the input (x, y, z) is a DH 3-tuple then all t output 3-tuples are random DH 3-tuples but with the middle term in common, and (2) when the input (x, y, z) is not a DH 3-tuple then $A_1, A_2, ..., A_t, B_1, R_1, R_2, ..., R_t \in_U G_{\mathfrak{p}}$ and $B_1 = B_2 = ... = B_t$.

$\texttt{DDHRerandN}((p,q), g, x, y, z, 2)$ is logically equivalent to algorithm DDHRerand5. To see this, note that the algorithm $\texttt{DDHRerandN}((p,q), g, x, y, z, 2)$ outputs the set of tuples $T = \{(A_1, B_1, R_1), (A_2, B_2, R_2)\}$ which, rearranging and dropping the B_2 yields the 5-tuple $(A_1, A_2, B_1, R_2, R_1)$. Observe that $B_1 = B_2$.

Let $\texttt{GetMiddle}(T)$ to be a function that on input a set T that is output by DDHRerandN, selects a tuple in T and returns the middle value in it. All middle values are the same so it doesn't matter which tuple is selected. We now address key anonymity for FBENCR.

Definition 8. If $\forall$ probabilistic polytime adversaries $\mathcal{A}$, $\forall\ \alpha > 0$, $\forall\ i \in \{1, 2, ..., w\}$, and $\forall$ sufficiently large κ, after the following,

1. generate $((p,q), g) \leftarrow \mathcal{IG}(\kappa)$
2. $((y_1, x_1), (y_2, x_2), ..., (y_w, x_w)) \leftarrow \texttt{FBGEN}((p,q), g)$
3. $(m_1, m_2, ..., m_w) \leftarrow \mathcal{A}((p,q), g, y_1, y_2, ..., y_w,$ "specify messages")
4. if $\exists\ j \in \{1, 2, ..., w\}$ such that $m_j \notin G_{\mathfrak{p}}$ then output "false" and halt
5. $(k_{j,0}, k_{j,1}) \in_U [1,q] \times [1,q]$ for $j = 1, 2, ..., w$
6. $\sigma_j \in_U \{1, 2, ..., w\}$ for $j = 1, 2, ..., w$
7. $(c_1, c_2, ..., c_w) \leftarrow \texttt{FBENCR}(p, g, (m_1, (k_{1,0}, k_{1,1}), \sigma_1), ..., (m_w, (k_{w,0}, k_{w,1}), \sigma_w), y_1, y_2, ..., y_w)$
8. $(\sigma'_1, \sigma'_2, ..., \sigma'_w) \leftarrow \mathcal{A}(c_1, c_2, ..., c_w,$"guess")
9. if $\sigma_i = \sigma'_i$ then output "true" else output "false"

the output of the experiment is "true" with probability less than $\frac{1}{w} + \frac{1}{\kappa^\alpha}$ then FBENCR is secure in the sense of key anonymity.

THEOREM 7. *If DDH is hard then algorithm* FBENCR *is secure in the sense of key anonymity.*

PROOF. Suppose there exists a probabilistic polynomial time adversary $\mathcal{A}$, an $\alpha > 0$, an $i \in \{1, 2, ..., w\}$, and a sufficiently large κ, such that $\mathcal{A}$ succeeds with probability greater than or equal to $\frac{1}{w} + \frac{1}{\kappa^\alpha}$. Consider algorithm AlgR9 that takes as input a DDH problem instance $((p,q), g, a_0, b_0, c_0)$.

$\texttt{AlgR9}((p,q), g, a_0, b_0, c_0)$:

1. $T_j \leftarrow \texttt{DDHRerandN}((p,q), g, a_0, b_0, c_0, 2w)$ for $j = 1, 2, ..., w$
2. set $y_j = \texttt{GetMiddle}(T_j)$ for $j = 1, 2, ..., w$
3. $(m_1, m_2, ..., m_w) \leftarrow \mathcal{A}((p,q), g, y_1, y_2, ..., y_w,$ "specify messages")
4. if $\exists\ j \in \{1, 2, ..., w\}$ such that $m_j \notin G_{\mathfrak{p}}$ then output "false" and halt
5. $\sigma_j \in_U \{1, 2, ..., w\}$ for $j = 1, 2, ..., w$
6. for j in 1..w do:
7. extract a tuple (A_0, B_0, R_0) without replacement from T_{σ_j}
8. extract a tuple (A_1, B_1, R_1) without replacement from T_{σ_j}
9. $c_j \leftarrow ((A_0, R_0), (A_1, R_1 m_j))$
10. $(\sigma'_1, \sigma'_2, ..., \sigma'_w) \leftarrow \mathcal{A}(c_1, c_2, ..., c_w,$"guess")
11. if $\sigma_i = \sigma'_i$ then output "true" else output "false"'

Consider the case that the input is a DH 3-tuple. It follows from the definition of DDHRerandN that c_j is a proper encryption of m_j using public key y_{σ_j} for $j = 1, 2, ..., w$ under FBENCR. Therefore, the input to $\mathcal{A}$ is drawn from the same set and probability distribution as the input to $\mathcal{A}$ in Definition 8. It follows that $\sigma_i = \sigma'_i$ with probability greater than or equal to $\frac{1}{w} + \frac{1}{\kappa^\alpha}$. So, for random exponents a and

b in $[1,q]$, $\Pr[\texttt{AlgR9}((p,q),g,g^a,g^b,g^{ab}) = \text{“true”}] \geq \frac{1}{w} + \frac{1}{\kappa^\alpha}$. Define $\psi = \Pr[\texttt{AlgR9}((p,q),g,g^a,g^b,g^{ab}) = \text{“true”}]$.

Now consider the case that the input is not a DH 3-tuple. It follows from the definition of DDHRerandN that c_j is uniformly distributed in $G_\mathfrak{p}^2 \times G_\mathfrak{p}^2$ and y_j is uniformly distributed in $G_\mathfrak{p}$ for $j = 1, 2, ..., w$. Let p_j be the probability that $\mathcal{A}$ responds with $\sigma'_i = j$ for $j = 1, 2, ..., w$. Then the probability that $\sigma_i = \sigma'_i$ is $\frac{1}{w}p_1 + \frac{1}{w}p_2 + ... + \frac{1}{w}p_w = \frac{1}{w}$. So, for randomly chosen exponents a, b, and c in $[1,q]$, $\Pr[\texttt{AlgR9}((p,q),g,g^a,g^b,g^c) = \text{“true”}] = \frac{q^2}{q^3}\psi + (1 - \frac{q^2}{q^3})\frac{1}{w}$ which is overwhelmingly close to $\frac{1}{w}$. □

We now address key anonymity for FBMIXER.

Definition 9. If $\forall$ probabilistic polytime adversaries $\mathcal{A}$, $\forall\ \alpha > 0$, $\forall\ i \in \{1, 2, ..., w\}$, and $\forall$ sufficiently large κ, after the following,

1. generate $((p,q),g) \leftarrow \mathcal{IG}(\kappa)$
2. $((y_1,x_1),(y_2,x_2),...,(y_w,x_w)) \leftarrow \texttt{FBGEN}((p,q),g)$
3. $((m_1,r_1,\sigma_1),(m_2,r_2,\sigma_2),...,(m_w,r_w,\sigma_w)) \leftarrow \mathcal{A}((p,q),g,$ $y_1,y_2,...,y_w,$“specify ciphertexts and receivers”)
4. if $\exists\ j \in \{1,2,...,w\}$ such that $m_j \notin G_\mathfrak{p}$ then output “false” and halt
5. if $\exists\ j \in \{1,2,...,w\}$ such that $r_j \notin [1,q] \times [1,q]$ then output “false” and halt
6. if $\exists\ j \in \{1,2,...,w\}$ such that $\sigma_j \notin S$ then output “false” and halt
7. $(c_1,c_2,...,c_w) \leftarrow \texttt{FBENCR}(p,g,(m_1,r_1,\sigma_1),...,$ $(m_w,r_w,\sigma_w),y_1,y_2,...,y_w)$
8. $\mu_j \in_U [1,q] \times [1,q]$ for $j = 1, 2, ..., w$
9. select a permutation π from S onto S uniformly at random
10. $(c'_{\pi(1)},c'_{\pi(2)},...,c'_{\pi(w)}) \leftarrow$ $\texttt{FBMIXER}(p,\pi,(c_1,\mu_1),(c_2,\mu_2),...,(c_w,\mu_w))$
11. $\pi' \leftarrow \mathcal{A}(c'_{\pi(1)},c'_{\pi(2)},...,c'_{\pi(w)},$“guess”)
12. if $\pi'(i) = \pi(i)$ then output “true” else output “false”

the output of the experiment is “true” with probability less than $\frac{1}{w} + \frac{1}{\kappa^\alpha}$ then FBMIXER is secure in the sense of anonymity.

THEOREM 8. *If DDH is hard then* FBMIXER *is secure in the sense of anonymity.*

PROOF. Suppose there exists a probabilistic polynomial time adversary $\mathcal{A}$, an $\alpha > 0$, an $i \in \{1, 2, ..., w\}$, and a sufficiently large κ, such that $\mathcal{A}$ succeeds with probability greater than or equal to $\frac{1}{w} + \frac{1}{\kappa^\alpha}$. Consider algorithm AlgR10 that takes as input a DDH problem instance $((p,q),g,a_0,b_0,c_0)$.

AlgR10$((p,q),g,a_0,b_0,c_0)$:
1. $T_j \leftarrow \texttt{DDHRerandN}((p,q),g,a_0,b_0,c_0,2w)$ for $j = 1, 2, ..., w$
2. set $y_j = \texttt{GetMiddle}(T_j)$ for $j = 1, 2, ..., w$
3. $((m_1,r_1,\sigma_1),(m_2,r_2,\sigma_2),...,(m_w,r_w,\sigma_w)) \leftarrow$ $\mathcal{A}((p,q),g,y_1,y_2,...,y_w,$ “specify ciphertexts and receivers”)
4. if $\exists\ j \in \{1,2,...,w\}$ such that $m_j \notin G_\mathfrak{p}$ then output “false” and halt
5. if $\exists\ j \in \{1,2,...,w\}$ such that $r_j \notin [1,q] \times [1,q]$ then output “false” and halt
6. if $\exists\ j \in \{1,2,...,w\}$ such that $\sigma_j \notin S$ then output “false” and halt
7. $(c_1,c_2,...,c_w) \leftarrow \texttt{FBENCR}(p,g,(m_1,r_1,\sigma_1),...,$ $(m_w,r_w,\sigma_w),y_1,y_2,...,y_w)$
8. for j in $1..w$ do:
9. extract a tuple (A_0,B_0,R_0) without replacement from T_{σ_j}
10. extract a tuple (A_1,B_1,R_1) without replacement from T_{σ_j}
11. $((\alpha_0,\beta_0),(\alpha_1,\beta_1)) \leftarrow c_j$
12. $c'_j \leftarrow ((\alpha_0 A_0,\beta_0 R_0),(\alpha_1 A_1,\beta_1 R_1))$
13. select a permutation π from S onto S uniformly at random
14. $(c'_{\pi(1)},c'_{\pi(2)},...,c'_{\pi(w)}) \leftarrow \texttt{fp}(\pi,c'_1,c'_2,...,c'_w)$
15. $\pi' \leftarrow \mathcal{A}(c'_{\pi(1)},c'_{\pi(2)},...,c'_{\pi(w)},$“guess”)
16. if $\pi'(i) = \pi(i)$ then output “true” else output “false”

Consider the case that the input is a DH 3-tuple. Clearly the ciphertexts $c_1, c_2, ..., c_w$ are as specified by $\mathcal{A}$. It follows from the definition of DDHRerandN that c'_j is a proper re-encryption of c_j under FBMIXER for $j = 1, 2, ..., w$. Therefore, the input to $\mathcal{A}$ is drawn from the same set and probability distribution as the input to $\mathcal{A}$ in Definition 9. It follows that $\pi'(i) = \pi(i)$ with probability greater than or equal to $\frac{1}{w} + \frac{1}{\kappa^\alpha}$. So, for random exponents a and b in $[1,q]$, $\Pr[\texttt{AlgR10}((p,q),g,g^a,g^b,g^{ab}) = \text{“true”}] \geq \frac{1}{w} + \frac{1}{\kappa^\alpha}$. Define the value $\psi = \Pr[\texttt{AlgR10}((p,q),g,g^a,g^b,g^{ab}) = \text{“true”}]$.

Now consider the case that the input is not a DH 3-tuple. It follows from the definition of DDHRerandN that y_j is uniformly distributed in $G_\mathfrak{p}$ for $j = 1, 2, ..., w$ and that c'_j is uniformly distributed in $G_\mathfrak{p}^2 \times G_\mathfrak{p}^2$ for $j = 1, 2, ..., w$. Let p_j be the probability that $\mathcal{A}$ responds with $\pi'(i) = j$ for $j = 1, 2, ..., w$. Then the probability that $\pi'(i) = \pi(i)$ is $\frac{1}{w}p_1 + \frac{1}{w}p_2 + ... + \frac{1}{w}p_w = \frac{1}{w}$. So, for randomly chosen exponents a, b, and c in $[1,q]$, $\Pr[\texttt{AlgR10}((p,q),g,g^a,g^b,g^c) =$ “true”$] = \frac{q^2}{q^3}\psi + (1 - \frac{q^2}{q^3})\frac{1}{w}$ which is overwhelmingly close to $\frac{1}{w}$. □

We now address message indistinguishability.

Definition 10. If $\forall$ probabilistic polytime adversaries $\mathcal{A}$, $\forall\ \alpha > 0$, $\forall\ i \in \{1, 2, ..., w\}$, and $\forall$ sufficiently large κ, after the following,

1. generate $((p,q),g) \leftarrow \mathcal{IG}(\kappa)$
2. $((y_1,x_1),(y_2,x_2),...,(y_w,x_w)) \leftarrow \texttt{FBGEN}((p,q),g)$
3. $((m_{1,0},m_{1,1},\sigma_1),(m_{2,0},m_{2,1},\sigma_2),...,(m_{w,0},m_{w,1},\sigma_w))$ $\leftarrow \mathcal{A}((p,q),g,y_1,y_2,...,y_w,$ “specify messages and receivers”)
4. if $\exists j \in \{1,2,...,w\}$ such that ($m_{j,0} \notin G_\mathfrak{p}$ or $m_{j,1} \notin G_\mathfrak{p}$) then output “false” and halt
5. if $\exists j \in \{1,2,...,w\}$ such that $m_{j,0} = m_{j,1}$ then output “false” and halt
6. if $\exists j \in \{1,2,...,w\}$ such that $\sigma_j \notin \{1,2,...,w\}$ then output “false” and halt
7. $(k_{j,0},k_{j,1}) \in_U [1,q] \times [1,q]$ for $j = 1, 2, ..., w$
8. $b_j \in_U \{0,1\}$ for $j = 1, 2, ..., w$
9. $(c_1,c_2,...,c_w) \leftarrow \texttt{FBENCR}(p,g,(m_{1,b_1},(k_{1,0},k_{1,1}),\sigma_1),...,$ $(m_{w,b_w},(k_{w,0},k_{w,1}),\sigma_w),y_1,y_2,...,y_w)$
10. $(b'_1,b'_2,...,b'_w) \leftarrow \mathcal{A}(c_1,c_2,...,c_w,$“guess”)
11. if $b_i = b'_i$ then output “true” else output “false”

the output of the experiment is “true” with probability less than $\frac{1}{2} + \frac{1}{\kappa^\alpha}$ then FBENCR is secure in the sense of message indistinguishability.

THEOREM 9. *If DDH is hard then* FBENCR *is secure in the sense of message indistinguishability.*

PROOF. Suppose there exists a probabilistic polynomial time adversary $\mathcal{A}$, an $\alpha > 0$, an $i \in \{1, 2, ..., w\}$, and a sufficiently large κ, such that $\mathcal{A}$ succeeds with probability greater than or equal to $\frac{1}{2}+\frac{1}{\kappa^\alpha}$. Consider algorithm AlgR7 that takes as input a DDH problem instance $((p, q), g, a_0, b_0, c_0)$.

AlgR7$((p, q), g, a_0, b_0, c_0)$:
1. $T_j \leftarrow$ DDHRerandN$((p, q), g, a_0, b_0, c_0, 2w)$ for $j = 1, 2, ..., w$
2. set $y_j =$ GetMiddle(T_j) for $j = 1, 2, ..., w$
3. $((m_{1,0}, m_{1,1}, \sigma_1), (m_{2,0}, m_{2,1}, \sigma_2), ..., (m_{w,0}, m_{w,1}, \sigma_w)) \leftarrow \mathcal{A}((p, q), g, y_1, y_2, ..., y_w,$ "specify messages and receivers")
4. if $\exists\ j \in \{1, 2, ..., w\}$ such that ($m_{j,0} \notin G_\mathfrak{p}$ or $m_{j,1} \notin G_\mathfrak{p}$) then output "false" and halt
5. if $\exists\ j \in \{1, 2, ..., w\}$ such that $m_{j,0} = m_{j,1}$ then output "false" and halt
6. if $\exists\ j \in \{1, 2, ..., w\}$ such that $\sigma_j \notin \{1, 2, ..., w\}$ then output "false" and halt
7. $b_j \in_U \{0, 1\}$ for $j = 1, 2, ..., w$
8. for j in $1..w$ do:
9. extract a tuple (A_0, B_0, R_0) without replacement from T_{σ_j}
10. extract a tuple (A_1, B_1, R_1) without replacement from T_{σ_j}
11. $c_j \leftarrow ((A_0, R_0), (A_1, R_1 m_{j,b_j}))$
12. $(b'_1, b'_2, ..., b'_w) \leftarrow \mathcal{A}(c_1, c_2, ..., c_w,$"guess")
13. if $b_i = b'_i$ then output "true" else output "false"'

Consider the case that the input is a DH 3-tuple. It follows from the definition of DDHRerandN that c_j is a proper encryption of m_{j,b_j} using public key y_{σ_j} for $j = 1, 2, ..., w$ under FBENCR. Therefore, the input to $\mathcal{A}$ is drawn from the same set and probability distribution as the input to $\mathcal{A}$ in Definition 10. It follows that $b_i = b'_i$ with probability greater than or equal to $\frac{1}{2} + \frac{1}{\kappa^\alpha}$. So, for random exponents a and b in $[1, q]$, $\Pr[\text{AlgR7}((p, q), g, g^a, g^b, g^{ab}) = \text{"true"}] \geq \frac{1}{2} + \frac{1}{\kappa^\alpha}$. Define $\psi = \Pr[\text{AlgR7}((p, q), g, g^a, g^b, g^{ab}) = \text{"true"}]$.

Now consider the case that the input is not a DH 3-tuple. It follows from the definition of DDHRerandN that c_j is uniformly distributed in $G_\mathfrak{p}^2 \times G_\mathfrak{p}^2$ and y_j is uniformly distributed in $G_\mathfrak{p}$ for $j = 1, 2, ..., w$. Let p_1 be the probability that $\mathcal{A}$ responds with $b'_i = 0$. Then the probability that $b_i = b'_i$ is $\frac{1}{2}p_1 + \frac{1}{2}(1-p_1) = \frac{1}{2}$. So, for randomly chosen exponents a, b, and c in $[1, q]$, the probability $\Pr[\text{AlgR7}((p, q), g, g^a, g^b, g^c) = \text{"true"}] = \frac{q^2}{q^3}\psi + (1 - \frac{q^2}{q^3})\frac{1}{2}$ which is overwhelmingly close to $\frac{1}{2}$. □

Definition 11. If $\forall$ probabilistic polytime adversaries $\mathcal{A}$, $\forall\ \alpha > 0$, $\forall\ i \in \{1, 2, ..., w\}$, and $\forall$ sufficiently large κ, after the following,

1. generate $((p, q), g) \leftarrow \mathcal{IG}(\kappa)$
2. $((y_1, x_1), (y_2, x_2), ..., (y_w, x_w)) \leftarrow$ FBGEN$((p, q), g)$
3. $(\pi, (m_{1,0}, m_{1,1}, r_{1,0}, r_{1,1}, \sigma_1), (m_{2,0}, m_{2,1}, r_{2,0}, r_{2,1}, \sigma_2), ..., (m_{w,0}, m_{w,1}, r_{w,0}, r_{w,1}, \sigma_w)) \leftarrow \mathcal{A}((p, q), g, y_1, y_2, ..., y_w,$"specify ciphertexts, receivers, and π")
4. if π is not a permutation from S onto S then output "false" and halt
5. if $\exists\ j \in \{1, 2, ..., w\}$ such that ($m_{j,0} \notin G_\mathfrak{p}$ or $m_{j,1} \notin G_\mathfrak{p}$) then output "false" and halt
6. if $\exists\ j \in \{1, 2, ..., w\}$ such that $m_{j,0} = m_{j,1}$ then output "false" and halt
7. if $\exists\ j \in \{1, 2, ..., w\}$ such that ($r_{j,0} \notin [1, q] \times [1, q]$ or $r_{j,1} \notin [1, q] \times [1, q]$) then output "false" and halt
8. if $\exists\ j \in \{1, 2, ..., w\}$ such that $\sigma_j \notin \{1, 2, ..., w\}$ then output "false" and halt
9. $b_j \in_U \{0, 1\}$ for $j = 1, 2, ..., w$
10. $(c_1, c_2, ..., c_w) \leftarrow$ FBENCR$(p, g, (m_{1,b_1}, r_{1,b_1}, \sigma_1), ..., (m_{w,b_w}, r_{w,b_w}, \sigma_w), y_1, y_2, ..., y_w)$
11. $r_j \in_U [1, q] \times [1, q]$ for $j = 1, 2, ..., w$
12. $(c'_{\pi(1)}, c'_{\pi(2)}, ..., c'_{\pi(w)}) \leftarrow$ FBMIXER$(p, \pi, (c_1, r_1), (c_2, r_2), ..., (c_w, r_w))$
13. $(c'_1, c'_2, ..., c'_w) \leftarrow$ fpinv$(\pi, c'_{\pi(1)}, c'_{\pi(2)}, ..., c'_{\pi(w)})$
14. $(b'_1, b'_2, ..., b'_w) \leftarrow \mathcal{A}(c'_1, c'_2, ..., c'_w,$"guess")
15. if $b_i = b'_i$ then output "true" else output "false"

the output of the experiment is "true" with probability less than $\frac{1}{2} + \frac{1}{\kappa^\alpha}$ then FBMIXER is secure in the sense of message indistinguishability.

THEOREM 10. *If DDH is hard then* FBMIXER *is secure in the sense of message indistinguishability.*

PROOF. Suppose there exists a probabilistic polynomial time adversary $\mathcal{A}$, an $\alpha > 0$, an $i \in \{1, 2, ..., w\}$, and a sufficiently large κ, such that $\mathcal{A}$ succeeds with probability greater than or equal to $\frac{1}{2}+\frac{1}{\kappa^\alpha}$. Consider algorithm AlgR8 that takes as input a DDH problem instance $((p, q), g, a_0, b_0, c_0)$.

AlgR8$((p, q), g, a_0, b_0, c_0)$:
1. $T_j \leftarrow$ DDHRerandN$((p, q), g, a_0, b_0, c_0, 2w)$ for $j = 1, 2, ..., w$
2. set $y_j =$ GetMiddle(T_j) for $j = 1, 2, ..., w$
3. $(\pi, (m_{1,0}, m_{1,1}, r_{1,0}, r_{1,1}, \sigma_1), (m_{2,0}, m_{2,1}, r_{2,0}, r_{2,1}, \sigma_2), ..., (m_{w,0}, m_{w,1}, r_{w,0}, r_{w,1}, \sigma_w)) \leftarrow \mathcal{A}((p, q), g, y_1, y_2, ..., y_w,$"specify ciphertexts, receivers, and π")
4. if π is not a permutation from S onto S then output "false" and halt
5. if $\exists\ j \in \{1, 2, ..., w\}$ such that ($m_{j,0} \notin G_\mathfrak{p}$ or $m_{j,1} \notin G_\mathfrak{p}$) then output "false" and halt
6. if $\exists\ j \in \{1, 2, ..., w\}$ such that $m_{j,0} = m_{j,1}$ then output "false" and halt
7. if $\exists\ j \in \{1, 2, ..., w\}$ such that ($r_{j,0} \notin [1, q] \times [1, q]$ or $r_{j,1} \notin [1, q] \times [1, q]$) then output "false" and halt
8. if $\exists\ j \in \{1, 2, ..., w\}$ such that $\sigma_j \notin \{1, 2, ..., w\}$ then output "false" and halt
9. $b_j \in_U \{0, 1\}$ for $j = 1, 2, ..., w$
10. $(c_1, c_2, ..., c_w) \leftarrow$ FBENCR$(p, g, (m_{1,b_1}, r_{1,b_1}, \sigma_1), ..., (m_{w,b_w}, r_{w,b_w}, \sigma_w), y_1, y_2, ..., y_w)$
11. for j in $1..w$ do:
12. extract a tuple (A_0, B_0, R_0) without replacement from T_{σ_j}
13. extract a tuple (A_1, B_1, R_1) without replacement from T_{σ_j}
14. $((\alpha_0, \beta_0), (\alpha_1, \beta_1)) \leftarrow c_j$
15. $c'_j \leftarrow ((\alpha_0 A_0, \beta_0 R_0), (\alpha_1 A_1, \beta_1 R_1))$
16. $(b'_1, b'_2, ..., b'_w) \leftarrow \mathcal{A}(c'_1, c'_2, ..., c'_w,$"guess")
17. if $b_i = b'_i$ then output "true" else output "false"

Consider the case that the input is a DH 3-tuple. Clearly the ciphertexts $c_1, c_2, ..., c_w$ are as specified by $\mathcal{A}$. It follows from the definition of DDHRerandN that c'_j is a proper

re-encryption of c_j under FBMIXER for $j = 1, 2, ..., w$. Therefore, the input to adversary $\mathcal{A}$ is drawn from the same set and probability distribution as the input to $\mathcal{A}$ in Definition 11. It follows that $b_i = b'_i$ with probability greater than or equal to $\frac{1}{2} + \frac{1}{\kappa^\alpha}$. So, for random exponents a and b in $[1, q]$, $\Pr[\texttt{AlgR8}((p, q), g, g^a, g^b, g^{ab}) = \text{“true”}] \geq \frac{1}{2} + \frac{1}{\kappa^\alpha}$. Define the value $\psi = \Pr[\texttt{AlgR8}((p, q), g, g^a, g^b, g^{ab}) = \text{“true”}]$.

Now consider the case that the input is not a DH 3-tuple. It follows from the definition of DDHRerandN that y_j is uniformly distributed in $G_\mathfrak{p}$ for $j = 1, 2, ..., w$ and that c'_j is uniformly distributed in $G_\mathfrak{p}^2 \times G_\mathfrak{p}^2$ for $j = 1, 2, ..., w$. Let p_1 be the probability that $\mathcal{A}$ responds with $b'_i = 0$. Then the probability that $b_i = b'_i$ is $\frac{1}{2}p_1 + \frac{1}{2}(1 - p_1) = \frac{1}{2}$. So, for randomly chosen exponents a, b, and c in $[1, q]$, the probability $\Pr[\texttt{AlgR8}((p, q), g, g^a, g^b, g^c) = \text{“true”}] = \frac{q^2}{q^3}\psi + (1 - \frac{q^2}{q^3})\frac{1}{2}$ which is overwhelmingly close to $\frac{1}{2}$. □

Theorems 7, 8, 9, and 10 show that properties 1, 2, 3, and 4 of a forward-anonymous batch mix hold, respectively. The FBMIXER servers store no keys at all so the forward-anonymity property holds (property 5). Theorem 8 proves that anonymity holds from a single honest mix. Therefore, the robustness property holds (property 6). Completeness and low-latency are straightforward to show (properties 7 and 8). Theorem 11 therefore holds.

THEOREM 11. *If DDH is hard then* FBMIX *is a forward-anonymous batch mix.*

10. CONCLUSION

We cryptanalyzed the security foundation of universal re-encryption and showed that the definition of security for it was insufficient. We presented a new definition of security for it that carefully handles message indistinguishability and key anonymity for both encryption and re-encryption. We also showed that the ElGamal based universal cryptosystem and incomparable cryptosystem were not proven secure under DDH as claimed. To address this we introduced a new DDH proof technique and used it to prove that they are secure under DDH. Finally, we presented a forward-anonymous batch mix secure under DDH.

11. REFERENCES

[1] M. Bellare, A. Boldyreva, A. Desai, and D. Pointcheval. Key-Privacy in Public-Key Encryption. In *Asiacrypt '01*, pages 566–582, 2001.

[2] D. Boneh. The Decision Diffie-Hellman Problem. In *Proceedings of the Third Algorithmic Number Theory Symposium*, pages 48–63, 1998.

[3] J. Camenisch and A. Lysyanskaya. A Formal Treatment of Onion Routing. In *Advances in Cryptology—Crypto '05*, pages 169–187, 2005.

[4] G. Danezis. Breaking Four Mix-related Schemes Based on Universal Re-encryption. *Int. J. Inf. Sec.*, 6(6):393–402, 2007.

[5] T. ElGamal. A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. In *Advances in Cryptology—Crypto '84*, pages 10–18, 1985.

[6] P. Fairbrother. An Improved Construction for Universal Re-encryption. In *Privacy Enhancing Technologies*, pages 79–87, 2004.

[7] S. Goldwasser and S. Micali. Probabilistic Encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[8] P. Golle. Reputable Mix Networks. In *Privacy Enhancing Technologies*, pages 51–62, 2004.

[9] P. Golle, M. Jakobsson, A. Juels, and P. Syverson. Universal Re-encryption for Mixnets. In *CT-RSA 2004*, pages 163–178, 2004.

[10] M. Gomułkiewicz, M. Klonowski, and M. Kutyłowski. Onions Based on Universal Re-encryption—Anonymous Communication Immune Against Repetitive Attack. In *WISA*, pages 400–410, 2004.

[11] J. Groth. Rerandomizable and Replayable Adaptive Chosen Ciphertext Attack Secure Cryptosystems. In *Theory of Cryptography—TCC '04*, pages 152–170, 2004.

[12] S. Hohenberger, G. N. Rothblum, A. Shelat, and V. Vaikuntanathan. Securely Obfuscating Re-encryption. *Journal of Cryptology*, 24(4):694–719, 2011.

[13] M. Klonowski, M. Kutyłowski, A. Lauks, and F. Zagórski. Universal Re-encryption of Signatures and Controlling Anonymous Information Flow. In *WARTACRYPT*, pages 179–188, 2004.

[14] M. Klonowski, M. Kutyłowski, and F. Zagórski. Anonymous Communication with On-line and Off-line Onion Encoding. In *SOFSEM*, pages 229–238, 2005.

[15] T. Lu, B. Fang, Y. Sun, and L. Guo. Some Remarks on Universal Re-encryption and a Novel Practical Anonymous Tunnel. In *ICCNMC*, pages 853–862, 2005.

[16] S. Micali, C. Rackoff, and B. Sloan. The Notion of Security for Probabilistic Cryptosystems. *SIAM Journal on Computing*, 17(2):412–426, 1988.

[17] M. Naor and O. Reingold. Number-Theoretic Constructions of Efficient Pseudo-Random Functions. In *IEEE FOCS '97*, pages 458–467, 1997.

[18] M. Prabhakaran and M. Rosulek. Rerandomizable RCCA Encryption. In *Advances in Cryptology—Crypto '07*, pages 517–534, 2007.

[19] M. Prabhakaran and M. Rosulek. Homomorphic Encryption with CCA Security. In *Automata, Languages and Programming*, pages 667–678, 2008.

[20] M. Stadler. Publicly Verifiable Secret Sharing. In *Advances in Cryptology—Eurocrypt '96*, pages 190–199, 1996.

[21] B. R. Waters, E. W. Felten, and A. Sahai. Receiver Anonymity via Incomparable Public Keys. In *ACM CCS*, pages 112–121, 2003.