U.C. Berkeley — CS276: Cryptography | Lecture 24
Professor: David Wagner | April 13, 2006
Scribe: Ilya Bagrak

Lecture 24

1 Mixnets

Suppose our task is to gather course feedback like HKN does at the end of the semester with an additional requirement that the students' feedback remains anonymous. In the pen and paper setup, the students trust HKN not to reveal their identity. In fact, it is generally assumed that no correspondence between the feedback and the identity of the student is leaked in the process. Can an analogous anonymous feedback system be implemented electronically?

Note that the problem is similar to the problem of maintaining voter anonymity in the context of electronic voting or anonymous email.

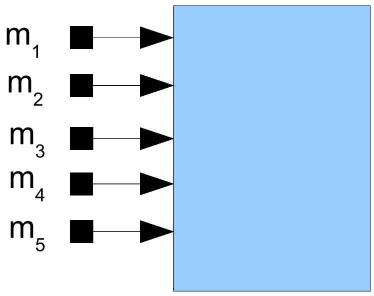

Figure 1: Naive mixnet

In general, the web is not anonymous. At the very least the eavesdropper can observe the traffic and correlate it with some originating IP, which can later be traced back to an individual user.

Assuming that we trust HKN, a simple solution is to use encryption. This setup is shown in Figure 2

It is easy to see that although encryption solves the privacy problem (i.e. an eavesdropper can no longer observe the contents of course surveys that go over the wire), an eavesdropper can still tell who is sending the message. We can remedy the situation by having HKN scramble the order in which the messages are output.

The resulting system (see Figure 3) is essentially a simplest functional mixnet. π denotes a permutation applied to the inputs of the mixnet to yield the outputs.

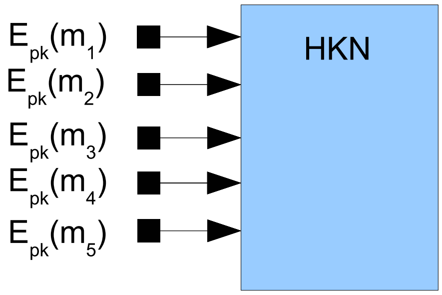

Figure 2: Encrypt-before-send Mixnet

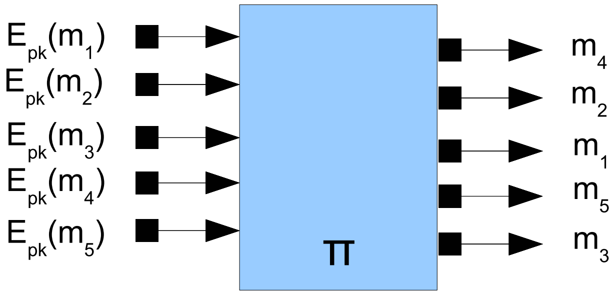

Figure 3: Permuting Mixnet

Note that we still trust HKN not to reveal the permutation π or modify contents of messages.

In the end, privacy and integrity are still predicated by trust in HKN. Can this trust be eliminated?

Our solution to maintaining privacy and integrity without trusting some intermediary can easily be applied anonymous remailers, voting, and anonymization of web traffic.

We have already seen a decryption mixnet (see Figure 3). An extension of this construct is the so-called blinding or re-encryption mixnet shown below.

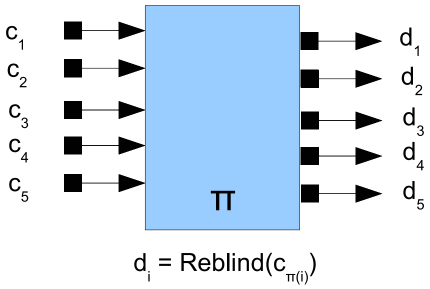

Figure 4: Blinding or Re-encryption Mixnet

We can now compose several blinding mixnets as seen in Figure 5.

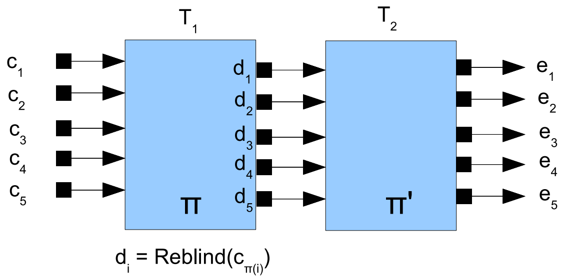

Figure 5: Mixnet Composition

It is self-evident that with one or more trusted mixnets is composed with any number of untrusted ones, the composition is trusted. Since the composition of mixnets corresponds to the composition of the permutations, $\pi' = \pi_n \circ \cdots \circ \pi_1$, if any π_i is random so is the resulting π'.

As always, we assume that all messages c_i have uniform length.

How do onlookers know that the mixnet servers really have permuted the original ciphertexts, without adding new ciphertexts, deleting old ones, or modifying any of the original ciphertexts? In general, the integrity goal is that the decryption of the ciphertexts at the input to the mixnet should yield the same (multi-)set of plaintexts as the decryption of the ciphertexts at the output to the mixnet. How can the mixnets prove that they have respected this integrity goal? One would would be for the mixnets to reveal all their secrets, as follows.

Theorem 1 (Mixnet Integrity)
A mixnet constructed as shown in Figure 4 can prove that it has respected the integrity goal.
Proof:[Proof sketch] The mixnet operator reveals:

1. the permutation π, and

2. the random coins used during reblinding.

The observer can then verify that $d_i = \text{Reblind}(c_{\pi(i)})$ for all i. □

The problem with the above protocol is that it involves revealing all the mixnet's secrets, destroying all anonymity and privacy (which, after all, was the whole purpose behind mixnets). Can we have both privacy and integrity? As we shall see in the next section, the answer is yes.

2 Zero-knowledge Proof of Mixnet Integrity

The proof shown above is of polynomial length and can be checked in polynomial time, so there must exist a corresponding zero-knowledge proof that does not require revealing the mixnet's secrets (since there is ZKIP for every language in NP). In practice, however, this generic solution is likely to have a running time so large as to make this approach impractical. Our goal then is to construct an efficient zero-knowledge interactive proof.

DEFINITION 1 ZKIP$(d, c) \equiv$ prove that d's are isomorphic to the c's (by analogy with ZKIP of graph isomorphism).

CLAIM 2 $(c = (c_1, \ldots, c_n) \simeq d = (d_1, \ldots, d_n))$
[Prover P]
$\sigma \xleftarrow{\$} Perm(n)$
$c'_i = Reblind(c_{\sigma(i)})$
Follow the protocol below...

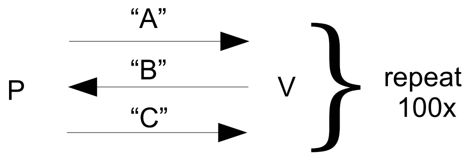

Figure 6: Zero-knowledge Interactive Proof of Mixnet Integrity

$$\begin{array}{lcl}
A & \equiv & c' = (c'_1, \ldots, c'_n) \\
B & \equiv & \text{challenge: } \begin{cases} \text{prove } c' \cong c & (i) \\ \text{prove } c' \cong d & (ii) \end{cases} \\
C & \equiv & \begin{cases} \sigma, & ZKIP(c'_i, c_{\sigma(i)}) \text{ for } i = 1 \ldots n \quad (i) \\ \sigma\pi^{-1}, & ZKIP(c'_i, d_{\sigma(\pi^{-1}(i))}) \text{ for } i = 1 \ldots n \quad (ii) \end{cases} \\
 & & \text{where } ZKIP(x, y) \text{ is a zero-knowledge interactive} \\
 & & \text{proof that } x = \text{Reblind}(y).
\end{array}$$

After the protocol is repeated 100 times, the probability that a prover is malicious, and he manages to convince the verifier of a falsehood is $(\frac{1}{2})^{100}$.

2.1 El Gamal

Consider a specific instance of a zero-knowledge interactive proof, where encryption is implemented with El Gamal.

DEFINITION 2 (EL GAMAL) Given $G = \langle g \rangle$ where $o(g) = q$ and q is prime, private key k, and public key $g^k = h$, $E(m) = (g^r, h^r * m)$ where $r \xleftarrow{\$} (\mathbb{Z}/p\mathbb{Z})$

We can now define a special instance of Reblind as

$$\text{Reblind}(\alpha, \beta) = (g^s * \alpha, h^s * \beta) = (g^{r+s}, h^{r+s} * m) = (\alpha', \beta')$$

where $s \xleftarrow{\$} (\mathbb{Z}/p\mathbb{Z})$.

The basic principle underlying this construction is that we get two ciphertexts, and we can't tell if they are from the same message, let alone learn anything about the message.

2.2 Non-interactive Zero Knowledge

Claim 3 ($(\alpha', \beta') = \text{Reblind}(\alpha, \beta)$)
Assuming an honest P and a random oracle model, we have the general construction of a zero-knowledge interactive *proof shown in Figure 7 (note that the challenge c is obtained by querying the random oracle $\mathcal{H}$ with prover's reblinding a):*

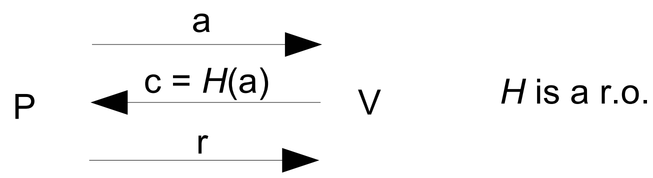

Figure 7: Zero-knowledge Interactive Proof with R.O.

We can extend this notion by thinking of a single message a containing a commitment to many ciphertexts, specifically a_1 through a_{200}.

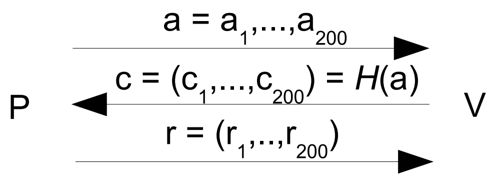

Figure 8: Zero-knowledge Interactive Proof with R.O.

In this setup, the probability that a prover chooses an a which gives exactly the challenge it wants is $Pr[\text{success}] = \frac{1}{2^{200}}$ (because $\mathcal{H}$ is a random function). After time t, the probability $Pr[\text{success}] = \frac{t}{2^{200}}$. As expected the probability of success is proportional to time t.

2.3 Fiat-Shamir Construction

Since both P and V contain oracle access to $\mathcal{H}$, we can construct a non-interactive version of the above proof such that the prover queries the oracle $\mathcal{H}$, generates the corresponding r.

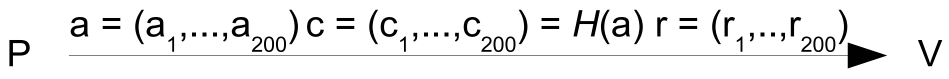

Figure 9: Fiat-Shamir ZKP Construction

Theorem 4 (ZKP in r.o. model)
It can be shown that every NP-language has a non-interactive zero knowledge proof in the r.o. model.

3 Visual Cryptography

At the end of the lecture Professor Wagner gave a demo of visual cryptography. The demo involved overlapping two transparencies, one being the key and the other the ciphertext, to produce a plaintext message.