BOISE STATE UNIVERSITY

Distributed blinding for distributed ElGamal re-encryption

Alejandro Anzola Ávila

October 16, 2019

Boise State University

Definitions

Motivation

This paper shows a protocol for interacting *distributed services* that emphasizes on step flexibility rather than evaluate on quantitative measures, such as number of messages exchanged or total computing time.

ElGamal public key encryption i

Is based on large prime numbers p and q such that

$$p = 2q + 1$$

Let $\mathcal{G}_p$ be a cyclic subgroup (of order q) of $\mathbb{Z}_p^* = \{i \mid 1 \leq i \leq p-1\}$, where g is a generator of $\mathcal{G}_p$.
Any $k \in \mathbb{Z}_q^*$ can be an ElGamal private key and $K = (p, q, g, y)$ is the public key, and $y = g^k \mod p$.

ElGamal public key encryption ii

An ElGamal ciphertext $E(m)$ for plaintext $m \in \mathcal{G}_p$ is a pair (g^r, my^r) with r uniformly and randomly chosen from $\mathbb{Z}_q^*$.
Ciphertext $E(m) = (a, b)$ is decrypted by computing b/a^k.

$$b/a^k = my^r/(g^r)^k = m(g^k)^r/(g^r)^k = m$$

When $E(m, r)$ is shown, the value of r is made explicit. Therefore $\mathcal{E}(m)$ is the set $\{E(m, r) \mid r \in \mathbb{Z}_q^*\}$ of all possible ciphertexts for m.

ElGamal public key encryption properties

Given $E(m_1) = (a_1, b_1)$, $E(m_2) = (a_2, b_2)$ and $E(m) = (a, b)$, we have

- $E(m)^{-1} = (a^{-1}, b^{-1})$
- $m' \cdot E(m) = (a, m', b)$
- $E(m_1) \cdot E(m_2) = (a_1 a_2, b_1 b_2)$

The following properties hold

ElGamal Inverse	$E(m)^{-1} \in \mathcal{E}(m^{-1})$
ElGamal Juxtaposition	$m' \cdot E(m, r) = E(m'm)$
ElGamal Multiplication[1]	If $r_1 + r_2 \in \mathbb{Z}_q^*$ then $E(m_1, r_1) \times E(m_2, r_2) \in \mathcal{E}(m_1 m_2)$

[1] Homomorphic property

Re-encryption and Distributed Blinding protocols

Re-encryption protocol

The basic re-encryption protocol is

1. Pick a random[2] $\rho \in \mathcal{G}_p$, then compute $E_A(\rho)$ and $E_B(\rho)$
2. Compute blinded ciphertext $E_A(m\rho) \coloneqq E_A(m) \times E_A(\rho)$
3. Employ threshold decryption to obtain blinded plaintext $m\rho$ from blinded ciphertext $E_A(m\rho)$.
4. Compute $E_B(m) \coloneqq m\rho \cdot E_B(\rho)^{-1}$

[2] The possibility of compromised servers makes computing ρ, $E_A(\rho)$, and $E_B(\rho)$ trickier.

Distributed blinding protocol i

Given two related public keys K_A and K_B, the distributed blinding protocol must satisfy the following correctness requirements:

- **Randomness-Confidentiality**: Blinding factor $\rho \in \mathcal{G}_p$ is chosen randomly and kept confidential from the adversary.
- **Consistency**: The protocol outputs a pair of ciphertexts $E_A(\rho)$ and $E_B(\rho)$.

They make an assumption about "Compromised servers":

Failstop adversaries
Compromised servers are limited to disclosing locally stored information or halting prematurely.

Distributed blinding protocol ii

To compute a confidential blinding factor ρ, its sufficient to calculate:

$$\prod_{i \in I} \rho_i$$

where I is the set of at least $f + 1$ servers. And each server $i \in I$ generates a random ρ_i.

Failstop adversary distributed blinding protocol i

1. Coordinator C_j initiates protocol by sending to every server in B an `init` message.
2. Upon receipt of an init message from C_j, a server i:
   2.1 Generates an independent random number ρ_i
   2.2 Computes encrypted contribution $(E_A(\rho_i), E_B(\rho))$
   2.3 Sends contribution to coordinator C_j
3. Upon receipt of contribute messages from a set I comprising $f+1$ servers in B
   3.1 C_j computes:
   $E_A(\rho) = \bigtimes_{i \in I} E_A(\rho_i)$
   $E_B(\rho) = \bigtimes_{i \in I} E_B(\rho_i)$
   3.2 Send $(E_A(\rho), E_B(\rho))$ to service A

$E_x(\rho) = \bigtimes_{i \in I} E_x(\rho_i, r_i)$ requires that $\sum_{i \in I} r_i \in \mathbb{Z}_q^*$ holds.

Failstop adversary distributed blinding protocol ii

To cope with faulty coordinators and guarantee protocol termination, $f + 1$ coordinators are used, that way at least one will complete the protocol.

Defending against malicious attacks

Defending against malicious attacks

Three forms of misbehaviour become possible:

1. Servers choose contributions that are not independent
2. The encrypted contribution from each server i **not** being of the form $(E_A(\rho_i), E_B(\rho_i'))$, where $\rho_i = \rho_i'$
3. Servers and coordinators not following the protocol in other ways

Randomness-Confidentiality

Problem

Suppose $\{(E_A(\rho_i), E_B(\rho_i)) \mid 1 \leq i \leq f\}$. After receiving these, a compromised server generates two ciphertexts $E_A(\hat{\rho}_i)$ and $E_B(\hat{\rho}_i)$ and constructs it in a way that does:

$$(E_A(\hat{\rho}_i) \times \left(\bigtimes_{i=1}^{f} E_A(\rho_i)\right)^{-1}, E_B(\hat{\rho}_i) \times \left(\bigtimes_{i=1}^{f} E_B(\rho_i)\right)^{-1})$$

Solution

Instead of sending an encrypted message to the coordinator, each server sends a *commitment*, which is a cryptographic hash of that encrypted contribution. And only after the coordinator has received $2f+1$ commitments does it solicit encrypted contributions from the servers. Then it needs to receive $f+1$ encrypted contributions.

Encrypted contribution consistency

Problem

The encrypted contribution from each server i **not** being of the form $(E_A(\rho_i), E_B(\rho'_i))$, where $\rho_i = \rho'_i$

Solution

They use a cryptographic block called *verifiable dual encryption* and it is based on the *non-interactive zero-knowledge proof*, for the equality of two discrete logarithms.