Introduction of the committed encryption

Weikeng Chen
for CS294-144 Blockchain course project

May 2018

1 Motivation

We want to encrypt one message to several individual parties. With public-key encryption under a group $\mathbb{G}$ and the group generator $g \in \mathbb{G}$, we assume there are three parties, and their public/private key pairs are (g^{x_1}, x_1), (g^{x_2}, x_2), and (g^{x_3}, x_3), respectively, we can encrypt a message m as follows:

$$\begin{aligned}
&r_1 \leftarrow_\$ [0, ..., |\mathbb{G}| - 1],\ r_2 \leftarrow_\$ [0, ..., |\mathbb{G}| - 1],\ r_3 \leftarrow_\$ [0, ..., |\mathbb{G}| - 1],\\
&k \leftarrow_\$ \{0,1\}^\lambda,\\
&c_1 = g^{r_1} \parallel \mathsf{KDF}(g^{x_1 r_1}) \oplus k,\\
&c_2 = g^{r_2} \parallel \mathsf{KDF}(g^{x_2 r_2}) \oplus k,\\
&c_3 = g^{r_3} \parallel \mathsf{KDF}(g^{x_3 r_3}) \oplus k,\\
&c_4 = \mathsf{AuthEnc.Enc}_k(\mathsf{msg}).
\end{aligned}$$

where $\mathsf{KDF}(\cdot)$ is a key derivation function and $\mathsf{AuthEnc}$ refers to an authenticated encryption scheme.

For honest encryption, all three parties can decrypt the message using their secret key. However, if the encryption is not honest, different parties may see different messages, we formalize as follows:

$$\begin{aligned}
&r_1 \leftarrow_\$ [0, ..., |\mathbb{G}| - 1],\ r_2 \leftarrow_\$ [0, ..., |\mathbb{G}| - 1],\ r_3 \leftarrow_\$ [0, ..., |\mathbb{G}| - 1],\\
&k \leftarrow_\$ \{0,1\}^\lambda,\\
&c_1 = g^{r_1} \parallel \mathsf{KDF}(g^{x_1 r_1}) \oplus k,\\
&c_2 = g^{r_2} \parallel \mathsf{KDF}(g^{x_2 r_2}) \oplus k \boxed{\oplus \Delta_2},\\
&c_3 = g^{r_3} \parallel \mathsf{KDF}(g^{x_3 r_3}) \oplus k \boxed{\oplus \Delta_3},\\
&c_4 = \mathsf{AuthEnc.Enc}_k(\mathsf{msg}).
\end{aligned}$$

in which the party 2 obtains $\mathsf{k} \oplus \Delta_2$ and the party 3 obtains $\mathsf{k} \oplus \Delta_3$ instead of k. Note that the party 2's key $\mathsf{k} \oplus \Delta_2$ may decrypt c_4 into non-$\perp$ result, as the authenticity of authenticated encryption does not necessarily protect against malicious encryption.

We want to provide a multi-recipient encryption that satisfies the following properties: (1) semantic security; (2) party 1 can detect whether c_2 and c_3 is generated honestly. Therefore, a malicious encryption cam be detected by the party 1 without knowing the secret keys of the party 2 and the party 3.

2 Our construction, informally

We encrypt the message as follows with a pseudorandom function:

$$\begin{aligned}
&s \leftarrow_{\$} \{0,1\}^\lambda,\ k = \mathsf{PRF}_s(1),\\
&r_1 \leftarrow_{\$} [0, ..., |\mathbb{G}| - 1],\\
&\qquad c_1 = g^{r_1} \,\|\, \mathsf{KDF}(g^{x_1 r_1}) \oplus s,\\
&r_2 = \mathsf{PRF}_s(2) \mod |\mathbb{G}|,\\
&\qquad c_2 = g^{r_2} \,\|\, \mathsf{KDF}(g^{x_2 r_2}) \oplus k,\\
&r_3 = \mathsf{PRF}_s(3) \mod |\mathbb{G}|,\\
&\qquad c_3 = g^{r_3} \,\|\, \mathsf{KDF}(g^{x_3 r_3}) \oplus k,\\
&c_4 = \mathsf{AuthEnc.Enc}_k(\mathsf{msg}).
\end{aligned}$$

assumed the Hash Diffie-Hellman (HDH) assumption holds on the group $\mathbb{G}$.

Semantic security. We can prove the semantic security of this construction with the random oracle heuristic.

By hybrid arguments, we have the following proof. We assume the distinguisher has all the public keys, but does not have any private keys of the three parties.

$\mathbf{H}_0$: the honest generation.

$\mathbf{H}_1$: in this hybrid, we change the s in c_1 into 0^λ:

$$\begin{aligned}
&s \leftarrow_{\$} \{0,1\}^\lambda,\ k = \mathsf{PRF}_s(1),\\
&r_1 \leftarrow_{\$} [0, ..., |\mathbb{G}| - 1],\\
&\qquad \boxed{c_1 = g^{r_1} \,\|\, \mathsf{KDF}(g^{x_1 r_1}),}\\
&r_2 = \mathsf{PRF}_s(2) \mod |\mathbb{G}|,\\
&\qquad c_2 = g^{r_2} \,\|\, \mathsf{KDF}(g^{x_2 r_2}) \oplus k,\\
&r_3 = \mathsf{PRF}_s(3) \mod |\mathbb{G}|,\\
&\qquad c_3 = g^{r_3} \,\|\, \mathsf{KDF}(g^{x_3 r_3}) \oplus k,\\
&c_4 = \mathsf{AuthEnc.Enc}_k(\mathsf{msg}).
\end{aligned}$$

This hybrid is computationally indistinguishable with $\mathbf{H}_0$ because $\mathsf{KDF}(g^{x_1 r_1})$ is computationally indistinguishable to a random λ-bit string assuming HDH.

$\mathbf{H}_2$: in this hybrid, we change k, r_2, and r_3 to randomly sampled values and discard s:

$$\begin{aligned}
&\boxed{k \leftarrow_{\$} \{0,1\}^\lambda,}\\
&r_1 \leftarrow_{\$} [0, ..., |\mathbb{G}| - 1],\\
&\qquad c_1 = g^{r_1} \,\|\, \mathsf{KDF}(g^{x_1 r_1}),\\
&\boxed{r_2 \leftarrow_{\$} [0, ..., |\mathbb{G}| - 1],}\\
&\qquad c_2 = g^{r_2} \,\|\, \mathsf{KDF}(g^{x_2 r_2}) \oplus k,\\
&\boxed{r_3 \leftarrow_{\$} [0, ..., |\mathbb{G}| - 1],}\\
&\qquad c_3 = g^{r_3} \,\|\, \mathsf{KDF}(g^{x_3 r_3}) \oplus k,\\
&c_4 = \mathsf{AuthEnc.Enc}_k(\mathsf{msg}).
\end{aligned}$$

This hybrid is computationally indistinguishable with $\mathbf{H}_1$ because of PRF with hidden seed s.

$\mathbf{H}_3$: in this hybrid, we hide the key k:

$$
\begin{aligned}
&r_1 \leftarrow_{\$} [0, ..., |\mathbb{G}| - 1], \\
&\quad c_1 = g^{r_1} \parallel \mathsf{KDF}(g^{x_1 r_1}), \\
&r_2 \leftarrow_{\$} [0, ..., |\mathbb{G}| - 1], \\
&\quad \boxed{c_2 = g^{r_2} \parallel \mathsf{KDF}(g^{x_2 r_2}),} \\
&r_3 \leftarrow_{\$} [0, ..., |\mathbb{G}| - 1], \\
&\quad \boxed{c_3 = g^{r_3} \parallel \mathsf{KDF}(g^{x_3 r_3}),} \\
&k \leftarrow_{\$} \{0,1\}^{\lambda}, \\
&c_4 = \mathsf{AuthEnc.Enc}_k(\mathsf{msg}).
\end{aligned}
$$

This hybrid is computationally indistinguishable with $\mathbf{H}_2$ due to the same reason for $\mathbf{H}_1 \overset{c}{\approx} \mathbf{H}_0$.

$\mathbf{H}_4$: in this hybrid, we change the message into 1^l, where l is the message size:

$$
\begin{aligned}
&r_1 \leftarrow_{\$} [0, ..., |\mathbb{G}| - 1], \\
&\quad c_1 = g^{r_1} \parallel \mathsf{KDF}(g^{x_1 r_1}), \\
&r_2 \leftarrow_{\$} [0, ..., |\mathbb{G}| - 1], \\
&\quad c_2 = g^{r_2} \parallel \mathsf{KDF}(g^{x_2 r_2}), \\
&r_3 \leftarrow_{\$} [0, ..., |\mathbb{G}| - 1], \\
&\quad c_3 = g^{r_3} \parallel \mathsf{KDF}(g^{x_3 r_3}), \\
&k \leftarrow_{\$} \{0,1\}^{\lambda}, \\
&c_4 = \mathsf{AuthEnc.Enc}_k(\boxed{1^l}).
\end{aligned}
$$

This hybrid is computationally indistinguishable with $\mathbf{H}_3$ due to the message indistinguishability of the authenticated encryption.

According to the hybrid arguments, $\mathbf{H}_0 \overset{c}{\approx} \mathbf{H}_4$, but $\mathbf{H}_4$ can be simulated by a PPT simulator without the message. This shows that our construction achieves semantic security under HDH assumption.

Accountability (in our definition). As mentioned in the motivation section, we want the party 1 to be able to detect malicious encryption, in which different parties will see different results.

We first introduce the problem. In a nutshell, we want to see if Δ_1, Δ_2, ..., and Δ_4 equal to zero pads and whether $k = k'$ in the following ciphertext:

$$
\begin{aligned}
&c_1 = g^{r_1} \parallel \mathsf{KDF}(g^{x_1 r_1}) \oplus s, \\
&c_2 = g^{r_2} \boxed{\oplus \Delta_1} \parallel \mathsf{KDF}(g^{x_2 r_2}) \oplus k \boxed{\oplus \Delta_2,} \\
&c_3 = g^{r_3} \boxed{\oplus \Delta_3} \parallel \mathsf{KDF}(g^{x_3 r_3}) \oplus k \boxed{\oplus \Delta_4,} \\
&c_4 = \mathsf{AuthEnc.Enc}_{k'}(\mathsf{msg}).
\end{aligned}
$$

where $k = \mathsf{PRF}_s(1)$, $r_2 = \mathsf{PRF}_s(2)$, and $r_3 = \mathsf{PRF}_s(3)$.

We now introduce the algorithm to detect the malicious encryption.

$$\mathsf{Detect}(c_1, c_2, c_3, c_4, \mathbb{G}, g, g^{x_1}, g^{x_2}, g^{x_3}, x_1)$$

which runs as follows:

1. Computes $g^{x_1 r_1}$ from g^{r_1} and x_1.
2. Recovers s by XOR the second part of c_1 with $\mathsf{KDF}(g^{x_1 r_1})$.
3. Computes the following pseudorandom values:
   - $k = \mathsf{PRF}_s(1)$.
   - $r_2 = \mathsf{PRF}_s(2)$.
   - $r_3 = \mathsf{PRF}_s(3)$.
4. Checks whether the first part of c_2 equals to g^{r_2} and the first part of c_3 equals to g^{r_3}. If not, outputs CHEAT and terminates. Otherwise, we know Δ_1 and Δ_3 are both zero pads, and the ciphertext has the following format:
$$\begin{aligned} c_1 &= g^{r_1} \parallel \mathsf{KDF}(g^{x_1 r_1}) \oplus s, \\ c_2 &= \boxed{g^{r_2}} \parallel \mathsf{KDF}(g^{x_2 r_2}) \oplus k \oplus \Delta_2, \\ c_3 &= \boxed{g^{r_3}} \parallel \mathsf{KDF}(g^{x_3 r_3}) \oplus k \oplus \Delta_4, \\ c_4 &= \mathsf{AuthEnc.Enc}_{k'}(\mathsf{msg}). \end{aligned}$$
5. Computes $g^{x_2 r_2}$ from r_2 and g^{x_2} and $g^{x_3 r_3}$ from r_3 and g^{x_3}. Note that we only use the public keys of the party 2 and the party 3.
6. Recovers Δ_2 by XOR the second part of c_2 with $\mathsf{KDF}(g^{x_2 r_2}) \oplus k$ and Δ_4 by XOR the second part of c_4 with $\mathsf{KDF}(g^{x_3 r_3}) \oplus k$.
7. Checks if Δ_2 and Δ_4 are both zero pads. If not, outputs CHEAT and terminates. Otherwise, we know the ciphertext has the following format:
$$\begin{aligned} c_1 &= g^{r_1} \parallel \mathsf{KDF}(g^{x_1 r_1}) \oplus s, \\ c_2 &= g^{r_2} \parallel \boxed{\mathsf{KDF}(g^{x_2 r_2}) \oplus k,} \\ c_3 &= g^{r_3} \parallel \boxed{\mathsf{KDF}(g^{x_3 r_3}) \oplus k,} \\ c_4 &= \mathsf{AuthEnc.Enc}_{k'}(\mathsf{msg}). \end{aligned}$$
8. Uses k to decrypt c_4. If the decrypted result is $\perp$, the ciphertext is not valid, the algorithm outputs CHEAT and terminates. Otherwise, outputs ACCEPT, as the ciphertext now has the following format the same as the one from honest encryption:
$$\begin{aligned} c_1 &= g^{r_1} \parallel \mathsf{KDF}(g^{x_1 r_1}) \oplus s, \\ c_2 &= g^{r_2} \parallel \mathsf{KDF}(g^{x_2 r_2}) \oplus k, \\ c_3 &= g^{r_3} \parallel \mathsf{KDF}(g^{x_3 r_3}) \oplus k, \\ c_4 &= \mathsf{AuthEnc.}\boxed{\mathsf{Enc}_k}(\mathsf{msg}). \end{aligned}$$

According to the discussion above, if the detection algorithm outputs ACCEPT, three parties will see the same decrypted result.