CS409 Report

Jay Mehta: 22B1281
Tanay Bhat: 22B3303

November 2024

Contents

Chapter 1

Introduction to Sigma Protocols

Zero-knowledge proofs are a type of cryptographic model that enable a prover to convince a verifier that a he knows a secret without revealing anything other than the claim itself. Sigma protocols provide a framework for realizing interactive zero-knowledge proofs in practice. Sigma protocols are essential components of blockchain systems, secure multi-party computation, and other cryptographic protocols. Hence, it is important to understand how sigma protocols work and how they can be used to build secure systems.

1.1 Definition

Before defining a sigma protocol, we need to define the notion of an effective relation.

Definition 1. *An effective relation is a relation $R \subseteq X \times Y$ is a binary relation such that X, Y, and R are all efficiently recognizable finite sets*[1]*. The element $x \in X$ is called the witness and $y \in Y$ is called the statement.*

We can now define a sigma protocol.

Definition 2. *Let $R \subseteq X \times Y$ be an effective relation and $(x, y) \in R$. An interactive two-party protocol specified by algorithms (P, V) is a sigma protocol for R with challenge space C and public input y if the following conditions hold:*

- ***Completeness**: For all $(x, y) \in R$, verifier V accepts with probability 1.*
- ***Special soundness**: There exists an efficient algorithm* EXT *which when given y and accepting transcripts (T, c_0, z_0) and (T, c_1, z_1) such that $c_0 \neq$*

[1] A set S is efficiently recognizable if there exists a polynomial time algorithm that can determine if a given element $x \in S$.

c_1, *outputs* x *such that* $(x, y) \in R$.[2] *Note that the first message is same in both transcripts.*

- ***Special honest verifier zero-knowledge****: There exists an efficient simulator* SIM *such that for all* $(y, c) \in Y \times C$, *the distribution of* $(T, z) = SIM(y, c)$ *is an accepting conversation for* y *and has the same distribution as the the transcript of conversation between* $P(x, y)$ *and* $V(y)$.

1.2 Sigma Protocol Flow

Here P is an interactive algorithm called the prover and accepts the pair $(x, y) \in R$ as input. The verifier V is an interactive algorithm that accepts a statement $y \in Y$ as input and outputs either accept or reject. A usual sigma protocol is carried out as follows:

1. The prover P computes as message t using (x, y) and sends it to the verifier V. This message is called the commitment.
2. The verifier V uniformly chooses a challenge $c \in C$ and sends it to P.
3. The prover P computes a response z using c and sends it to V.
4. The verifier V outputs accept or reject based on the response z. This output is computed using the statement y and the conversation (t, c, z).

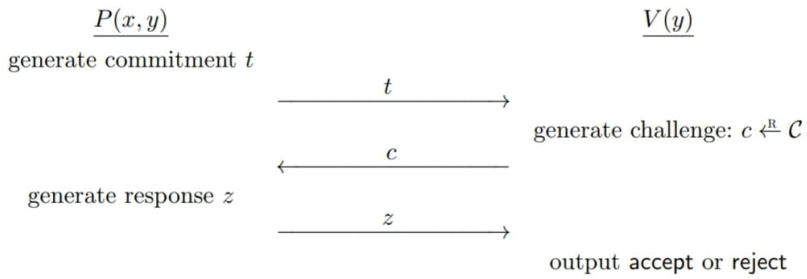

Figure 1.1: Sigma Protocol Flow

The completeness property ensures that a prover who knows the secret can always convince the verifier. The soundness property ensures that the prover cannot fake knowing the witness. The zero-knowledge property ensures that the verifier learns nothing about the witness other than the fact that the statement is true if the verfier follows the protocol honestly. We shall now show how one can combine sigma protocols using the AND and OR composition techniques.

[2]This property may be generalized to k-special soundness for $k > 2$ for higher security applications.

Chapter 2

AND Composition

We shall assume that the base sigma protocols for both the AND and OR composition are valid and satisfy the completeness, soundness, and zero-knowledge properties.

2.1 Construction

Suppose we have two sigma protocols (P_0, V_0) and (P_1, V_1) for relations $R_0 \subseteq X_0 \times Y_0$ and $R_1 \subseteq X_0 \times Y_0$ respectively. We can construct a new sigma protocol for the following relation:

$$R = \{(x_0, x_1, y_0, y_1) \in X_0 \times X_1 \times Y_0 \times Y_1 \mid (x_0, y_0) \in R_0 \wedge (x_1, y_1) \in R_1\} \quad (2.1)$$

Hence given a statement $y_0 \in Y_0$ and $y_1 \in Y_1$, the prover can prove that he knows x_0 and x_1 to the verifier. The construction of the AND composition is as follows:

1. The prover P receives (x_0, x_1, y_0, y_1) as input and computes the commitments t_0 and t_1 using (x_0, y_0) and (x_1, y_1) respectively. The prover sends (t_0, t_1) to V.
2. The verifier V uniformly chooses a challenge $c \in C$ and sends it to P.
3. P computes the responses z_0 and z_1 using c on (x_0, y_0) and (x_1, y_1) respectively. The prover sends (z_0, z_1) to V.
4. V outputs accept if both (t_0, c, z_0) and (t_1, c, z_1) are accepted by the verifier V_0 and V_1 respectively. Otherwise, V outputs reject.

We shall now show that this is infact a valid sigma protocol by proving the completeness, soundness, and zero-knowledge properties.

2.2 Proof of Completeness

Theorem 1. *The AND composition of two sigma protocols is complete.*

Proof. Let $(x_0, y_0) \in R_0$ and $(x_1, y_1) \in R_1$. We know that by completeness of (P_0, V_0) and (P_1, V_1), the verifier V_0 and V_1 will accept these inputs with probability 1. Hence by construction, the verifier V will accept the inputs (x_0, x_1, y_0, y_1) with probability 1. This shows tha the AND composition is complete. □

2.3 Proof of Soundness

Theorem 2. *The AND composition of two sigma protocols satisfies special soundness.*

Proof. Let $((t_0, t_1), c_0, (z_0, z_1))$ and $((t_0, t_1), c_0', (z_0', z_1'))$ be accepting transcripts for the AND composition and $(y_0, y_1) \in Y_0 \times Y_1$ be the input statement. Note that $(t_0, c_0, z_0), (t_0, c_0', z_0')$ and $(t_1, c_1, z_1), (t_1, c_1', z_1')$ are accepting transcripts for (P_0, V_0) for input y_0 and (P_1, V_1) for input y_1 respectively based on the construction of the AND composition. As (P_0, V_0) and (P_1, V_1) satisfy special soundness, we know that there exist polynomial time algorithms EXT_0 and EXT_1 that can extract the witnesses x_0 and x_1 using the accepting transcripts and the imputs. Using this we construct an algorithm EXT as follows:

1. Given (y_0, y_1) and the accepting transcripts for the AND composition $((t_0, t_1), c_0, (z_0, z_1))$ and $((t_0, t_1), c_0', (z_0', z_1'))$, run EXT_0 on (y_0, c_0, z_0) and (y_0, c_0', z_0') to get x_0.
2. Run EXT_1 on (y_1, c_1, z_1) and (y_1, c_1', z_1') to get x_1.
3. Return (x_0, x_1).

Note that EXT runs in polynomial time. We can show this as follows:

$$\begin{aligned} T_{EXT} &= T_{EXT_0} + T_{EXT_1} \\ &= O(\text{poly}_0(n)) + O(\text{poly}_1(n)) \\ &\leq m_0 n^{d_0} + m_1 n^{d_1} \\ &\leq (m_0 + m_1) n^{\max(d_0, d_1)} \end{aligned} \tag{2.2}$$

Where T_{EXT} is the running time of EXT, T_{EXT_0} and T_{EXT_1} are the running times of EXT_0 and EXT_1 respectively, n is the input size, m_0 and m_1 are constants and d_0 and d_1 are the degrees of the polynomials. Hence EXT runs in polynomial time. We also know that EXT must output an accepting (x_0, x_1) as EXt_0 and EXT_1 are valid extractors for their respective protocols. This shows that the AND composition satisfies special soundness. □

2.4 Proof of Zero-Knowledge

Theorem 3. *The AND composition of two sigma protocols satisfies special honest verifier zero-knowledge.*

Proof. As (P_0, V_0) and (P_1, V_1) are valid sigma protocols and satisfy special honest verifier zero-knowledge, we know that there exist polynomial time simulators SIM_0 and SIM_1 that can simulate accepting transcripts for (P_0, V_0) and (P_1, V_1) respectively. We can construct a simulator SIM for the AND composition that accepts $((y_0, y_1), c) \in (Y_0 \times Y_1) \times C$ as follows:

1. Run SIM_0 on (y_0, c) to get (t_0, z_0).
2. Run SIM_1 on (y_1, c) to get (t_1, z_1).
3. Return $((t_0, t_1), (z_0, z_1))$.

□

Note that SIM_0 and SIM_1 run in polynomial time and can generate accepting transcripts for (P_0, V_0) and (P_1, V_1) respectively. The runtime of SIM is as follows:

$$\begin{aligned} T_{SIM} &= T_{SIM_0} + T_{SIM_1} \\ &= O(\text{poly}_0(n)) + O(\text{poly}_1(n)) \\ &\leq m_0 n^{d_0} + m_1 n^{d_1} \\ &\leq (m_0 + m_1) n^{\max(d_0, d_1)} \end{aligned} \tag{2.3}$$

Where T_{SIM} is the running time of SIM, T_{SIM_0} and T_{SIM_1} are the running times of SIM_0 and SIM_1 respectively, n is the input size, m_0 and m_1 are constants and d_0 and d_1 are the degrees of the polynomials. Hence SIM runs in polynomial time. As (t_0, c, z_0) and (t_1, c, z_1) are accepting transcripts for (P_0, V_0) and (P_1, V_1) respectively, from the construction of the AND composition, we know that $((t_0, t_1), c, (z_0, z_1))$ is an accepting transcript for the AND composition. To show that the simulator output has the same distribution as the accepting transcript, we see that:

$$\begin{aligned} (t_0, c, z_0) &\sim \langle P_0, V_0 \rangle (x_0, y_0) \\ (t_1, c, z_1) &\sim \langle P_1, V_1 \rangle (x_1, y_1) \\ ((t_0, t_1), c, (z_0, z_1)) &\sim \langle P, V \rangle ((x_0, x_1), (y_0, y_1)) \end{aligned} \tag{2.4}$$

The first two equations are by the zero-knowledge property of (P_0, V_0) and (P_1, V_1) respectively. The third equation is by the construction of the AND composition. Hence the AND composition satisfies special honest verifier zero-knowledge.

Chapter 3

OR Composition

3.1 Construction

Suppose we have two sigma protocols (P_0, V_0) and (P_1, V_1) for relations $R_0 \subseteq X_0 \times Y_0$ and $R_1 \subseteq X_0 \times Y_0$ respectively. We can construct a new sigma protocol for the following relation:

$$R = \{(b, x, y_0, y_1) \in \{0,1\} \times (X_0 \cup X_1) \times Y_0 \times Y_1 \mid (x, y_b) \in R_b\} \tag{3.1}$$

Hence given a statement $y_0 \in Y_0$ and $y_1 \in Y_1$, the prover can prove that he knows x such that $(x, y_0) \in R_0$ or $(x, y_1) \in R_1$ to the verifier. The verifier does not know whether the prover has a witness for R_0 or R_1. The construction of the OR composition is as follows:

1. The prover P receives (b, x, y_0, y_1) as input and initialize $d = 1 - b$. P uniformly chooses $c_d \in C$ and computes (t_d, z_d) using $SIM_d(y_d, c_d)$ where SIM_d is the simulator for (P_d, V_d).
2. P runs P_b on (x, y_b) to get t_b and sends (t_0, t_1) to V.
3. The verifier V uniformly chooses a challenge $c \in C$ and sends it to P.
4. P computes $c_b = c \oplus c_d$ and z_b using c_b on $P_b(x, y_b)$. The prover sends (c_0, z_0, z_1) to V.
5. V computes $c_1 = c \oplus c_0$ and checks if (t_0, c_0, z_0) and (t_1, c_1, z_1) are accepting transcripts for (P_0, V_0) or (P_1, V_1) respectively. Output accept if and only if both of them are accepted.

3.2 Proof of Completeness

Theorem 4. *The OR composition of two sigma protocols is complete.*

Proof. We shall show completeness for the input of form $(0, x, y_0, y_1) \in R$. The proof for $b = 1$ can be seen by the symmetry of the construction. The input implies that $(x, y_0) \in R_0$ and $d = 1$. Lets walk through the protocol:

1. P first chooses c_1 uniformly at random from the challenge space and (t_1, z_1) is computed using $SIM_1(y_1, c_1)$. Note that (t_1, z_1) is an accepting transcript for (P_1, V_1) for input y_1.
2. P_0 is run on (x, y_0) to get t_0.
3. P receives c from V and sets $c_0 = c \oplus c_1$. z_0 is computed using c_0 on $P_0(x, y_0)$. Note that (t_0, c_0, z_0) is an accepting transcript for (P_0, V_0) for input y_0. P sends (c_0, z_0, z_1) to V.
4. V computes $c_1 = c \oplus c_0 = c \oplus (c \oplus c_1) = c_1$. V checks if (t_0, c_0, z_0) and (t_1, c_1, z_1) are accepting transcripts for (P_0, V_0) and (P_1, V_1) respectively. As (t_0, c_0, z_0) is an accepting transcript for (P_0, V_0) and (t_1, c_1, z_1) is an accepting transcript for (P_1, V_1), V outputs accept.

Hence the OR composition is complete. □

3.3 Proof of Soundness

Theorem 5. *The OR composition of two sigma protocols satisfies special soundness.*

Proof. Let EXT_0 and EXT_1 be extractors for (P_0, V_0) and (P_1, V_1) respectively. Let $((t_0, t_1), c, (c_0, z_0, z_1))$ and $((t_0, t_1), c', (c'_0, z'_0, z'_1))$ be accepting transcripts for the OR composition. Let $(y_0, y_1) \in Y_0 \times Y_1$ be the input statement. We can compute $c_1 = c \oplus c_0$ and $c'_1 = c' \oplus c'_0$ and by the constructuion of the OR composition, we know that $(t_0, c_0, z_0), (t_0, c'_0, z'_0)$ and $(t_1, c_1, z_1), (t_1, c'_1, z'_1)$ are accepting transcripts for (P_0, V_0) and (P_1, V_1) respectively. We can now construct an extractor EXT as follows:

$$EXT \Rightarrow \begin{cases} \text{return } (0, EXT_0(y_0, (t_0, c_0, z_0), (t_0, c'_0, z'_0))), & \text{if } c_0 \neq c'_0 \\ \text{return } (1, EXT_1(y_1, (t_1, c_1, z_1), (t_1, c'_1, z'_1))), & \text{if } c_1 \neq c'_1 \end{cases} \tag{3.2}$$

Given that $c \neq c'$, we know that $c_0 \neq c'_0$ or $c_1 \neq c'_1$ must hold. Also note that EXT runs in polynomial time:

$$\begin{aligned} T_{EXT} &= \max(T_{EXT_0}, T_{EXT_1}) \\ &= \max(O(\text{poly}_0(n)), O(\text{poly}_1(n)) \\ &\leq \max(m_0 n^{d_0}, m_1 n^{d_1}) \\ &\leq \max(m_0, m_1) n^{\max(d_0, d_1)} \end{aligned} \tag{3.3}$$

We cann also see that EXT generates accepting transcripts. If $c_0 \neq c_0'$ then EXT outputs $(0, x_0)$ where $(x_0, y_0) \in R_0$. Hence $(0, x_0, y_0, y_1) \in R$. If $c_1 \neq c_1'$ then EXT outputs $(1, x_1)$ where $(x_1, y_1) \in R_1$. Hence $(1, x_1, y_0, y_1) \in R$. This shows that the OR composition satisfies special soundness.

□

3.4 Proof of Zero-Knowledge

Theorem 6. *The OR composition of two sigma protocols satisfies special honest verifier zero-knowledge.*

Proof. Given simulators SIM_0 and SIM_1 for (P_0, V_0) and (P_1, V_1) respectively, we can construct a simulator SIM for the OR composition as follows:

$$SIM((y_0, y_1), c) = ((t_0, t_1), (c_0, z_0, z_1)) \tag{3.4}$$

Where c_0 is drawn uniformly from the challenge space and $c_1 = c \oplus c_0$ is computed. We then compute $(t_0, z_0) = SIM_0(y_0, c_0)$ and $(t_1, z_1) = SIM_1(y_1, c_1)$. The runtime of SIM is as follows:

$$\begin{aligned} T_{SIM} &= T_{SIM_0} + T_{SIM_1} \\ &= O(\text{poly}_0(n)) + O(\text{poly}_1(n)) \\ &\leq m_0 n^{d_0} + m_1 n^{d_1} \\ &\leq (m_0 + m_1) n^{\max(d_0, d_1)} \end{aligned} \tag{3.5}$$

Hence SIM runs in polynomial time. We know that (t_0, c_0, z_0) and (t_1, c_1, z_1) have the same distribution as the accepting transcripts for (P_0, V_0) and (P_1, V_1) respectively. Hence by the construction of the OR composition, the transcript $((t_0, t_1), c, (c_0, z_0, z_1))$ is accepted for the OR composition. The similarity in distribution holds as the base sigma protocols satisfy special honest verifier zero-knowledge. Hence the OR composition satisfies special honest verifier zero-knowledge. □