Group Signatures

Łukasz Jeż

12 grudnia 2005

Własności

Definicja

Schematem podpisów grupowych nazwiemy schemat podpisów o następujących własnościach:

- *wiadomości mogą podpisać wyłącznie członkowie grupy*
- *„odbiorca podpisu” może sprawdzić, że jest to podpis danej grupy, ale nie jest w stanie stwierdzić, który jej członek jest autorem podpisu*
- *w razie czego, podpis może zostać „otwarty”, by poznać autora podpisu*

Własności

Definicja

Schematem podpisów grupowych nazwiemy schemat podpisów o następujących własnościach:

- *wiadomości mogą podpisać wyłącznie członkowie grupy*
- *„odbiorca podpisu” może sprawdzić, że jest to podpis danej grupy, ale nie jest w stanie stwierdzić, który jej członek jest autorem podpisu*
- *w razie czego, podpis może zostać „otwarty”, by poznać autora podpisu*

Własności

Definicja

Schematem podpisów grupowych nazwiemy schemat podpisów o następujących własnościach:

- *wiadomości mogą podpisać wyłącznie członkowie grupy*
- *„odbiorca podpisu” może sprawdzić, że jest to podpis danej grupy, ale nie jest w stanie stwierdzić, który jej członek jest autorem podpisu*
- *w razie czego, podpis może zostać „otwarty”, by poznać autora podpisu*

Własności

Definicja

Schematem podpisów grupowych nazwiemy schemat podpisów o następujących własnościach:

- *wiadomości mogą podpisać wyłącznie członkowie grupy*
- *„odbiorca podpisu” może sprawdzić, że jest to podpis danej grupy, ale nie jest w stanie stwierdzić, który jej członek jest autorem podpisu*
- *w razie czego, podpis może zostać „otwarty”, by poznać autora podpisu*

Przykłady zastosowań

- wydruki w dużej firmie, posiadającej wydziały
- przetargi (niepubliczne?)
- podpisywanie dokumentów (umów?) w imieniu (dużej) firmy
- pieniądze elektroniczne

Przykłady zastosowań

- wydruki w dużej firmie, posiadającej wydziały
- przetargi (niepubliczne?)
- podpisywanie dokumentów (umów?) w imieniu (dużej) firmy
- pieniądze elektroniczne

Przykłady zastosowań

- wydruki w dużej firmie, posiadającej wydziały
- przetargi (niepubliczne?)
- podpisywanie dokumentów (umów?) w imieniu (dużej) firmy
- pieniądze elektroniczne

Przykłady zastosowań

- wydruki w dużej firmie, posiadającej wydziały
- przetargi (niepubliczne?)
- podpisywanie dokumentów (umów?) w imieniu (dużej) firmy
- pieniądze elektroniczne

Przykłady zastosowań

- wydruki w dużej firmie, posiadającej wydziały
- przetargi (niepubliczne?)
- podpisywanie dokumentów (umów?) w imieniu (dużej) firmy
- pieniądze elektroniczne

Najprostsze rozwiązanie (1)

1. TA wybiera dowolny schemat podpisów z kluczem publicznym.
2. TA generuje i wręcza każdemu listę kluczy prywatnych.
3. TA publikuje w losowej kolejności pełną listę kluczy publicznych.
4. Każdy może podpisać wiadomość używając klucza ze swojej listy.
5. Każdego klucza można użyć tylko raz!

TA może podszyć się pod dowolnego członka grupy!

Najprostsze rozwiązanie (1)

1. TA wybiera dowolny schemat podpisów z kluczem publicznym.
2. TA generuje i wręcza każdemu listę kluczy prywatnych.
3. TA publikuje w losowej kolejności pełną listę kluczy publicznych.
4. Każdy może podpisać wiadomość używając klucza ze swojej listy.
5. Każdego klucza można użyć tylko raz!

TA może podszyć się pod dowolnego członka grupy!

Najprostsze rozwiązanie (1)

1. TA wybiera dowolny schemat podpisów z kluczem publicznym.
2. TA generuje i wręcza każdemu listę kluczy prywatnych.
3. TA publikuje w losowej kolejności pełną listę kluczy publicznych.
4. Każdy może podpisać wiadomość używając klucza ze swojej listy.
5. Każdego klucza można użyć tylko raz!

TA może podszyć się pod dowolnego członka grupy!

Najprostsze rozwiązanie (1)

1. TA wybiera dowolny schemat podpisów z kluczem publicznym.
2. TA generuje i wręcza każdemu listę kluczy prywatnych.
3. TA publikuje w losowej kolejności pełną listę kluczy publicznych.
4. Każdy może podpisać wiadomość używając klucza ze swojej listy.
5. Każdego klucza można użyć tylko raz!

TA może podszyć się pod dowolnego członka grupy!

Najprostsze rozwiązanie (1)

1. TA wybiera dowolny schemat podpisów z kluczem publicznym.
2. TA generuje i wręcza każdemu listę kluczy prywatnych.
3. TA publikuje w losowej kolejności pełną listę kluczy publicznych.
4. Każdy może podpisać wiadomość używając klucza ze swojej listy.
5. Każdego klucza można użyć tylko raz!

TA może podszyć się pod dowolnego członka grupy!

Najprostsze rozwiązanie (1)

1. TA wybiera dowolny schemat podpisów z kluczem publicznym.
2. TA generuje i wręcza każdemu listę kluczy prywatnych.
3. TA publikuje w losowej kolejności pełną listę kluczy publicznych.
4. Każdy może podpisać wiadomość używając klucza ze swojej listy.
5. Każdego klucza można użyć tylko raz!

TA może podszyć się pod dowolnego członka grupy!

Najprostsze rozwiązanie (1)

1. TA wybiera dowolny schemat podpisów z kluczem publicznym.
2. TA generuje i wręcza każdemu listę kluczy prywatnych.
3. TA publikuje w losowej kolejności pełną listę kluczy publicznych.
4. Każdy może podpisać wiadomość używając klucza ze swojej listy.
5. Każdego klucza można użyć tylko raz!

TA może podszyć się pod dowolnego członka grupy!

Schemat podpisów ElGamala - przypomnienie

- publiczne: p – l. pierwsza, $\alpha \in \mathbb{Z}_p^*$, $\beta \equiv \alpha^a \mod p$
- prywatne: a
- podpisem pod x jest $sig(x, k) = (\gamma, \delta)$, gdzie
  - $k \in \mathbb{Z}_{p-1}^*$ jest losowane i tajne
  - $\gamma = \alpha^k \mod p$
  - $\delta = (x - a\gamma)k^{-1} \mod (p-1)$
- weryfikacja: $\beta^\gamma \gamma^\delta \stackrel{?}{\equiv} \alpha^x \mod p$

Poprawienie (1)

1. Niech g będzie generatorem $\mathbb{Z}_p^*$, p – l. pierwsza
2. Każdy członek grupy, i, tworzy swój klucz prywatny s_i, a g^{s_i}, klucz publiczny, przesyła do TA
3. „Co tydzień” TA wręcza członkowi i losową liczbę $r_i \in \{1, \ldots, p-1\}$ i publikuje listę „zaślepionych” kluczy publicznych $(g^{s_i})^{r_i}$.
4. W tym tygodniu członek i używa $s_i r_i \mod (p-1)$ jako klucza prywatnego.

Poprawienie (1)

1. Niech g będzie generatorem $\mathbb{Z}_p^*$, p – l. pierwsza
2. Każdy członek grupy, i, tworzy swój klucz prywatny s_i, a g^{s_i}, klucz publiczny, przesyła do TA
3. „Co tydzień” TA wręcza członkowi i losową liczbę $r_i \in \{1, \ldots, p-1\}$ i publikuję listę „zaślepionych” kluczy publicznych $(g^{s_i})^{r_i}$.
4. W tym tygodniu członek i używa $s_i r_i \mod (p-1)$ jako klucza prywatnego.

Poprawienie (1)

1. Niech g będzie generatorem $\mathbb{Z}_p^*$, p – l. pierwsza
2. Każdy członek grupy, i, tworzy swój klucz prywatny s_i, a g^{s_i}, klucz publiczny, przesyła do TA
3. „Co tydzień” TA wręcza członkowi i losową liczbę $r_i \in \{1, \ldots, p-1\}$ i publikuję listę „zaślepionych” kluczy publicznych $(g^{s_i})^{r_i}$.
4. W tym tygodniu członek i używa $s_i r_i \mod (p-1)$ jako klucza prywatnego.

Poprawienie (1)

1. Niech g będzie generatorem $\mathbb{Z}_p^*$, p – l. pierwsza
2. Każdy członek grupy, i, tworzy swój klucz prywatny s_i, a g^{s_i}, klucz publiczny, przesyła do TA
3. „Co tydzień” TA wręcza członkowi i losową liczbę $r_i \in \{1, \ldots, p-1\}$ i publikuje listę „zaślepionych” kluczy publicznych $(g^{s_i})^{r_i}$.
4. W tym tygodniu członek i używa $s_i r_i \mod (p-1)$ jako klucza prywatnego.

Wymagania

Schemat musi spełniać poniższe warunki:

- niepodrabialność podpisów
- anonimowość podpisów
- niemożność powiązania podpisów
- brak wrabiania
- brak fałszywych oskarżeń

Wymagania

Schemat musi spełniać poniższe warunki:

- niepodrabialność podpisów
- anonimowość podpisów
- niemożność powiązania podpisów
- brak wrabiania
- brak fałszywych oskarżeń

Wymagania

Schemat musi spełniać poniższe warunki:

- niepodrabialność podpisów
- anonimowość podpisów
- niemożność powiązania podpisów
- brak wrabiania
- brak fałszywych oskarżeń

Wymagania

Schemat musi spełniać poniższe warunki:

- niepodrabialność podpisów
- anonimowość podpisów
- niemożność powiązania podpisów
- brak wrabiania
- brak fałszywych oskarżeń

Wymagania

Schemat musi spełniać poniższe warunki:

- niepodrabialność podpisów
- anonimowość podpisów
- niemożność powiązania podpisów
- brak wrabiania
- brak fałszywych oskarżeń

Wymagania

Schemat musi spełniać poniższe warunki:

- niepodrabialność podpisów
- anonimowość podpisów
- niemożność powiązania podpisów
- brak wrabiania
- brak fałszywych oskarżeń

Na co będziemy zwracać uwagę

- Czy TA (Trusted Authotity) jest potrzebne wyłącznie do ustanowienia grupy, czy również później? (np. do otwarcia podpisu)
- Czy moc TA można rozdzielić pomiędzy więcej stron?
- Czy do grupy mogą dołączać kolejne osoby? (a jak z będzie z usuwaniem?)
- rodzaj podpisów i założenia gwarantujące bezpieczeństwo
- wszelkiego rodzaju koszty...

Na co będziemy zwracać uwagę

- Czy TA (Trusted Authotity) jest potrzebne wyłącznie do ustanowienia grupy, czy również później? (np. do otwarcia podpisu)
- Czy moc TA można rozdzielić pomiędzy więcej stron?
- Czy do grupy mogą dołączać kolejne osoby? (a jak z będzie z usuwaniem?)
- rodzaj podpisów i założenia gwarantujące bezpieczeństwo
- wszelkiego rodzaju koszty...

Na co będziemy zwracać uwagę

- Czy TA (Trusted Authotity) jest potrzebne wyłącznie do ustanowienia grupy, czy również później? (np. do otwarcia podpisu)
- Czy moc TA można rozdzielić pomiędzy więcej stron?
- Czy do grupy mogą dołączać kolejne osoby? (a jak z będzie z usuwaniem?)
- rodzaj podpisów i założenia gwarantujące bezpieczeństwo
- wszelkiego rodzaju koszty...

Na co będziemy zwracać uwagę

- Czy TA (Trusted Authotity) jest potrzebne wyłącznie do ustanowienia grupy, czy również później? (np. do otwarcia podpisu)
- Czy moc TA można rozdzielić pomiędzy więcej stron?
- Czy do grupy mogą dołączać kolejne osoby? (a jak z będzie z usuwaniem?)
- rodzaj podpisów i założenia gwarantujące bezpieczeństwo
- wszelkiego rodzaju koszty...

Na co będziemy zwracać uwagę

- Czy TA (Trusted Authotity) jest potrzebne wyłącznie do ustanowienia grupy, czy również później? (np. do otwarcia podpisu)
- Czy moc TA można rozdzielić pomiędzy więcej stron?
- Czy do grupy mogą dołączać kolejne osoby? (a jak z będzie z usuwaniem?)
- rodzaj podpisów i założenia gwarantujące bezpieczeństwo
- wszelkiego rodzaju koszty...

Na co będziemy zwracać uwagę

- Czy TA (Trusted Authotity) jest potrzebne wyłącznie do ustanowienia grupy, czy również później? (np. do otwarcia podpisu)
- Czy moc TA można rozdzielić pomiędzy więcej stron?
- Czy do grupy mogą dołączać kolejne osoby? (a jak z będzie z usuwaniem?)
- rodzaj podpisów i założenia gwarantujące bezpieczeństwo
- wszelkiego rodzaju koszty...

Koszt

Interesują nas zależności poniższych wartości od liczby członków grupy; czasem także od liczby wiadomości, jaka ma zostać podpisana.

- długość klucza publicznego grupy
- ilość obliczeń w trakcie protokołu potwierdzenia podpisu
- ilość przesłanych bitów podczas protokołu potwierdzenia podpisu
- (ewentualnie) te same wielkości podczas protokołu wyparcia się podpisu

Koszt

Interesują nas zależności poniższych wartości od liczby członków grupy; czasem także od liczby wiadomości, jaka ma zostać podpisana.

- długość klucza publicznego grupy
- ilość obliczeń w trakcie protokołu potwierdzenia podpisu
- ilość przesłanych bitów podczas protokołu potwierdzenia podpisu
- (ewentualnie) te same wielkości podczas protokołu wyparcia się podpisu

Koszt

Interesują nas zależności poniższych wartości od liczby członków grupy; czasem także od liczby wiadomości, jaka ma zostać podpisana.

- długość klucza publicznego grupy
- ilość obliczeń w trakcie protokołu potwierdzenia podpisu
- ilość przesłanych bitów podczas protokołu potwierdzenia podpisu
- (ewentualnie) te same wielkości podczas protokołu wyparcia się podpisu

Koszt

Interesują nas zależności poniższych wartości od liczby członków grupy; czasem także od liczby wiadomości, jaka ma zostać podpisana.

- długość klucza publicznego grupy
- ilość obliczeń w trakcie protokołu potwierdzenia podpisu
- ilość przesłanych bitów podczas protokołu potwierdzenia podpisu
- (ewentualnie) te same wielkości podczas protokołu wyparcia się podpisu

Założenia dot. bezpieczeństwa

Przypuszczenie (1)

trudność problemu RSA roots – pociąga za sobą trudność problemu faktoryzacji oraz trudność problemu logarytmu dyskretnego modulo duża liczba złożona

Przypuszczenie (2)

trudność problemu logarytmu dyskretnego modulo duża liczba pierwsza

Założenia dot. bezpieczeństwa

Przypuszczenie (1)

trudność problemu RSA roots – pociąga za sobą trudność problemu faktoryzacji oraz trudność problemu logarytmu dyskretnego modulo duża liczba złożona

Przypuszczenie (2)

trudność problemu logarytmu dyskretnego modulo duża liczba pierwsza

Założenia dot. bezpieczeństwa

Przypuszczenie (1)

trudność problemu RSA roots – pociąga za sobą trudność problemu faktoryzacji oraz trudność problemu logarytmu dyskretnego modulo duża liczba złożona

Przypuszczenie (2)

trudność problemu logarytmu dyskretnego modulo duża liczba pierwsza

Założenia dot. bezpieczeństwa

Przypuszczenie (1)

trudność problemu RSA roots – pociąga za sobą trudność problemu faktoryzacji oraz trudność problemu logarytmu dyskretnego modulo duża liczba złożona

Przypuszczenie (2)

trudność problemu logarytmu dyskretnego modulo duża liczba pierwsza

Schematy oparte na podpisach niezaprzeczalnych i protokole Fiata-Shamira

- Wyobraźmy sobie rozszerzenie podpisów niezaprzeczalnych na grupę.
- Członek grupy będzie dowodził, że to on podpisał wiadomość.
- Ale w taki sposób, by ujawnić tylko tyle, że należy do grupy.
- Stosowane protokoły powinny być dowodami z (obliczeniową) wiedzą zerową.

Schematy oparte na podpisach niezaprzeczalnych i protokole Fiata-Shamira

- Wyobraźmy sobie rozszerzenie podpisów niezaprzeczalnych na grupę.
- Członek grupy będzie dowodził, że to on podpisał wiadomość.
- Ale w taki sposób, by ujawnić tylko tyle, że należy do grupy.
- Stosowane protokoły powinny być dowodami z (obliczeniową) wiedzą zerową.

Schematy oparte na podpisach niezaprzeczalnych i protokole Fiata-Shamira

- Wyobraźmy sobie rozszerzenie podpisów niezaprzeczalnych na grupę.
- Członek grupy będzie dowodził, że to on podpisał wiadomość.
- Ale w taki sposób, by ujawnić tylko tyle, że należy do grupy.
- Stosowane protokoły powinny być dowodami z (obliczeniową) wiedzą zerową.

Schematy oparte na podpisach niezaprzeczalnych i protokole Fiata-Shamira

- Wyobraźmy sobie rozszerzenie podpisów niezaprzeczalnych na grupę.
- Członek grupy będzie dowodził, że to on podpisał wiadomość.
- Ale w taki sposób, by ujawnić tylko tyle, że należy do grupy.
- Stosowane protokoły powinny być dowodami z (obliczeniową) wiedzą zerową.

Schematy oparte na podpisach niezaprzeczalnych i protokole Fiata-Shamira

- Wyobraźmy sobie rozszerzenie podpisów niezaprzeczalnych na grupę.
- Członek grupy będzie dowodził, że to on podpisał wiadomość.
- Ale w taki sposób, by ujawnić tylko tyle, że należy do grupy.
- Stosowane protokoły powinny być dowodami z (obliczeniową) wiedzą zerową.

Schemat (2)

1. TA wybiera l. pierwsze p i q oraz funkcję jednokierunkową f
2. TA wręcza członkowi i klucz prywatny s_i – losowo wybraną liczbą pierwszą ze zbioru $\Phi = \left\{\lceil\sqrt{N}\rceil, \ldots, \lfloor 2\sqrt{N}\rfloor - 1\right\}$
3. TA publikuje $N = pq$, $v = \prod s_i$ oraz f
4. Podpisem członka i pod wiadomością n będzie $(f(n))^{s_i} \bmod N$

Po k-krotnym powtórzeniu poniższego protokołu $\mathcal{V}$ będzie przekonany (z ppb $1 - 2^{-k}$), że $c \in \widetilde{\Omega} = \{\alpha - \beta, \ldots, \alpha + 2\beta\}$.

Schemat (2)

1. TA wybiera l. pierwsze p i q oraz funkcję jednokierunkową f
2. TA wręcza członkowi i klucz prywatny s_i – losowo wybraną liczbą pierwszą ze zbioru $\Phi = \left\{\lceil\sqrt{N}\rceil, \ldots, \lfloor 2\sqrt{N}\rfloor - 1\right\}$
3. TA publikuje $N = pq$, $v = \prod s_i$ oraz f
4. Podpisem członka i pod wiadomością n będzie $(f(n))^{s_i} \bmod N$

Po k-krotnym powtórzeniu poniższego protokołu $\mathcal{V}$ będzie przekonany (z ppb $1 - 2^{-k}$), że $c \in \widetilde{\Omega} = \{\alpha - \beta, \ldots, \alpha + 2\beta\}$.

Schemat (2)

1. TA wybiera l. pierwsze p i q oraz funkcję jednokierunkową f
2. TA wręcza członkowi i klucz prywatny s_i – losowo wybraną liczbą pierwszą ze zbioru $\Phi = \left\{\lceil\sqrt{N}\rceil, \ldots, \lfloor 2\sqrt{N}\rfloor - 1\right\}$
3. TA publikuje $N = pq$, $v = \prod s_i$ oraz f
4. Podpisem członka i pod wiadomością n będzie $(f(n))^{s_i} \bmod N$

Po k-krotnym powtórzeniu poniższego protokołu $\mathcal{V}$ będzie przekonany (z ppb $1 - 2^{-k}$), że $c \in \widetilde{\Omega} = \{\alpha - \beta, \ldots, \alpha + 2\beta\}$.

Schemat (2)

1. TA wybiera l. pierwsze p i q oraz funkcję jednokierunkową f
2. TA wręcza członkowi i klucz prywatny s_i – losowo wybraną liczbą pierwszą ze zbioru $\Phi = \left\{\lceil\sqrt{N}\rceil, \ldots, \lfloor 2\sqrt{N}\rfloor - 1\right\}$
3. TA publikuje $N = pq$, $v = \prod s_i$ oraz f
4. Podpisem członka i pod wiadomością n będzie $(f(n))^{s_i} \bmod N$

Po k-krotnym powtórzeniu poniższego protokołu $\mathcal{V}$ będzie przekonany (z ppb $1 - 2^{-k}$), że $c \in \widetilde{\Omega} = \{\alpha - \beta, \ldots, \alpha + 2\beta\}$.

Schemat (2)

1. TA wybiera l. pierwsze p i q oraz funkcję jednokierunkową f
2. TA wręcza członkowi i klucz prywatny s_i – losowo wybraną liczbą pierwszą ze zbioru $\Phi = \left\{\lceil\sqrt{N}\rceil, \ldots, \lfloor 2\sqrt{N}\rfloor - 1\right\}$
3. TA publikuje $N = pq$, $v = \prod s_i$ oraz f
4. Podpisem członka i pod wiadomością n będzie $(f(n))^{s_i} \bmod N$

Po k-krotnym powtórzeniu poniższego protokołu $\mathcal{V}$ będzie przekonany (z ppb $1 - 2^{-k}$), że $c \in \widetilde{\Omega} = \{\alpha - \beta, \ldots, \alpha + 2\beta\}$.

Schemat (2)

1. TA wybiera l. pierwsze p i q oraz funkcję jednokierunkową f
2. TA wręcza członkowi i klucz prywatny s_i – losowo wybraną liczbą pierwszą ze zbioru $\Phi = \left\{\lceil\sqrt{N}\rceil, \ldots, \lfloor 2\sqrt{N}\rfloor - 1\right\}$
3. TA publikuje $N = pq$, $v = \prod s_i$ oraz f
4. Podpisem członka i pod wiadomością n będzie $(f(n))^{s_i} \bmod N$

Po k-krotnym powtórzeniu poniższego protokołu $\mathcal{V}$ będzie przekonany (z ppb $1 - 2^{-k}$), że $c \in \widetilde{\Omega} = \{\alpha - \beta, \ldots, \alpha + 2\beta\}$.

Czego dowodzi osoba podpisująca

Osoba podpisująca wiadomość dowodzi znajomości s takiego, że

- $S \equiv m^s \mod N$
- $s \in \Phi$
- $s | v$

Gdzie publicznie znane są: N, v, $m = f(N)$, S, Φ; $m, S \in \mathbb{Z}_N^*$

Czego dowodzi osoba podpisująca

Osoba podpisująca wiadomość dowodzi znajomości s takiego, że

- $S \equiv m^s \mod N$
- $s \in \Phi$
- $s | v$

Gdzie publicznie znane są: N, v, $m = f(N)$, S, Φ; $m, S \in \mathbb{Z}_N^*$

Czego dowodzi osoba podpisująca

Osoba podpisująca wiadomość dowodzi znajomości s takiego, że

- $S \equiv m^s \mod N$
- $s \in \Phi$
- $s | v$

Gdzie publicznie znane są: N, v, $m = f(N)$, S, Φ; $m, S \in \mathbb{Z}_N^*$

Czego dowodzi osoba podpisująca

Osoba podpisująca wiadomość dowodzi znajomości s takiego, że

- $S \equiv m^s \mod N$
- $s \in \Phi$
- $s | v$

Gdzie publicznie znane są: N, v, $m = f(N)$, S, Φ; $m, S \in \mathbb{Z}_N^*$

Czego dowodzi osoba podpisująca

Osoba podpisująca wiadomość dowodzi znajomości s takiego, że

- $S \equiv m^s \mod N$
- $s \in \Phi$
- $s | v$

Gdzie publicznie znane są: N, v, $m = f(N)$, S, Φ; $m, S \in \mathbb{Z}_N^*$

Protokół potwierdzenia dla schematu (2)

Protokół (1)

sekret $\mathcal{P}$	:	c
publiczne	:	N, x, y, Ω; $x, y \in \mathbb{Z}_N^*, \Omega = \{\alpha, \ldots, \alpha + \beta\} \subset \mathbb{N}$
$\mathcal{P}$ *dowodzi* $\mathcal{V}$	:	$x^c \equiv y \mod N \wedge c \in \Omega$

Protokół potwierdzenia dla schematu (2)

Protokół (1)

1. $\mathcal{P}$ *wybiera* $r \in \{0, \ldots, \beta\}$. *Oblicza* $\mathcal{B}(z_1 \equiv x^r \mod N)$ *i* $\mathcal{B}(z_2 \equiv x^{r-\beta} \mod N)$ *i wysyła* $\{\mathcal{B}(z_1), \mathcal{B}(z_2)\}$ *do* $\mathcal{V}$.[a]
2. $\mathcal{V}$ *losuje* $b \in \{0, 1\}$ *i wysyła do* $\mathcal{P}$.
3. $\mathcal{P}$ *przesyła* $\mathcal{V}$ *w zależności od* b
   - $b = 0$ r *i ujawnia* z_1 *i* z_2
   - $b = 1$ $\widetilde{r} = (c + r)$ *bądź* $c + r - \beta$ *– element z* Ω *i ujawnia odpowiednio* z_1 *lub* z_2, *oznaczane przez* $\widetilde{z}$
4. $\mathcal{V}$ *sprawdza, że*
   - $b = 0$ $r \in \{0, \ldots, \beta\}$ *i że* $\mathcal{B}$ *zawierały* x^r *i* $x^{r-\beta}$
   - $b = 1$ $\widetilde{r} \in \Omega$ *oraz że jedno z* $\mathcal{B}$ *zawierało* $\widetilde{z}$ *takie, że* $x^{\widetilde{r}} \equiv \widetilde{z}y$

[a] $\mathcal{B}$ to "computationally secure blob"

Protokół potwierdzenia dla schematu (2)

Protokół (1)

1. $\mathcal{P}$ *wybiera* $r \in \{0, \ldots, \beta\}$. *Oblicza* $\mathcal{B}(z_1 \equiv x^r \mod N)$ *i* $\mathcal{B}(z_2 \equiv x^{r-\beta} \mod N)$ *i wysyła* $\{\mathcal{B}(z_1), \mathcal{B}(z_2)\}$ *do* $\mathcal{V}$.[a]
2. $\mathcal{V}$ *losuje* $b \in \{0, 1\}$ *i wysyła do* $\mathcal{P}$.
3. $\mathcal{P}$ *przesyła* $\mathcal{V}$ *w zależności od* b
   - $b = 0$ r *i ujawnia* z_1 *i* z_2
   - $b = 1$ $\widetilde{r} = (c + r)$ *bądź* $c + r - \beta$ *– element z* Ω *i ujawnia odpowiednio* z_1 *lub* z_2, *oznaczane przez* $\widetilde{z}$
4. $\mathcal{V}$ *sprawdza, że*
   - $b = 0$ $r \in \{0, \ldots, \beta\}$ *i że* $\mathcal{B}$ *zawierały* x^r *i* $x^{r-\beta}$
   - $b = 1$ $\widetilde{r} \in \Omega$ *oraz że jedno z* $\mathcal{B}$ *zawierało* $\widetilde{z}$ *takie, że* $x^{\widetilde{r}} \equiv \widetilde{z}y$

[a] $\mathcal{B}$ to "computationally secure blob"

Protokół potwierdzenia dla schematu (2)

Protokół (1)

1. $\mathcal{P}$ *wybiera* $r \in \{0, \ldots, \beta\}$. *Oblicza* $\mathcal{B}(z_1 \equiv x^r \mod N)$ *i* $\mathcal{B}(z_2 \equiv x^{r-\beta} \mod N)$ *i wysyła* $\{\mathcal{B}(z_1), \mathcal{B}(z_2)\}$ *do* $\mathcal{V}$.[a]
2. $\mathcal{V}$ *losuje* $b \in \{0, 1\}$ *i wysyła do* $\mathcal{P}$.
3. $\mathcal{P}$ *przesyła* $\mathcal{V}$ *w zależności od* b
   - $b = 0$ r *i ujawnia* z_1 *i* z_2
   - $b = 1$ $\widetilde{r} = (c + r)$ *bądź* $c + r - \beta$ *– element z* Ω *i ujawnia odpowiednio* z_1 *lub* z_2, *oznaczane przez* $\widetilde{z}$
4. $\mathcal{V}$ *sprawdza, że*
   - $b = 0$ $r \in \{0, \ldots, \beta\}$ *i że* $\mathcal{B}$ *zawierały* x^r *i* $x^{r-\beta}$
   - $b = 1$ $\widetilde{r} \in \Omega$ *oraz że jedno z* $\mathcal{B}$ *zawierało* $\widetilde{z}$ *takie, że* $x^{\widetilde{r}} \equiv \widetilde{z}y$

[a] $\mathcal{B}$ to "computationally secure blob"

Protokół potwierdzenia dla schematu (2)

Protokół (1)

1. $\mathcal{P}$ *wybiera* $r \in \{0, \ldots, \beta\}$. *Oblicza* $\mathcal{B}(z_1 \equiv x^r \mod N)$ *i* $\mathcal{B}(z_2 \equiv x^{r-\beta} \mod N)$ *i wysyła* $\{\mathcal{B}(z_1), \mathcal{B}(z_2)\}$ *do* $\mathcal{V}$.[a]
2. $\mathcal{V}$ *losuje* $b \in \{0, 1\}$ *i wysyła do* $\mathcal{P}$.
3. $\mathcal{P}$ *przesyła* $\mathcal{V}$ *w zależności od* b
   - $b = 0$ r *i ujawnia* z_1 *i* z_2
   - $b = 1$ $\widetilde{r} = (c + r)$ *bądź* $c + r - \beta$ *– element z* Ω *i ujawnia odpowiednio* z_1 *lub* z_2*, oznaczane przez* $\widetilde{z}$
4. $\mathcal{V}$ *sprawdza, że*
   - $b = 0$ $r \in \{0, \ldots, \beta\}$ *i że* $\mathcal{B}$ *zawierały* x^r *i* $x^{r-\beta}$
   - $b = 1$ $\widetilde{r} \in \Omega$ *oraz że jedno z* $\mathcal{B}$ *zawierało* $\widetilde{z}$ *takie, że* $x^{\widetilde{r}} \equiv \widetilde{z}y$

[a] $\mathcal{B}$ to "computationally secure blob"

Protokół potwierdzenia dla schematu (2)

Protokół (1)

1. $\mathcal{P}$ *wybiera* $r \in \{0, \ldots, \beta\}$. *Oblicza* $\mathcal{B}(z_1 \equiv x^r \mod N)$ *i* $\mathcal{B}(z_2 \equiv x^{r-\beta} \mod N)$ *i wysyła* $\{\mathcal{B}(z_1), \mathcal{B}(z_2)\}$ *do* $\mathcal{V}$.[a]
2. $\mathcal{V}$ *losuje* $b \in \{0, 1\}$ *i wysyła do* $\mathcal{P}$.
3. $\mathcal{P}$ *przesyła* $\mathcal{V}$ *w zależności od* b
   - $b = 0$ r *i ujawnia* z_1 *i* z_2
   - $b = 1$ $\widetilde{r} = (c + r)$ *bądź* $c + r - \beta$ *– element z* Ω *i ujawnia odpowiednio* z_1 *lub* z_2*, oznaczane przez* $\widetilde{z}$
4. $\mathcal{V}$ *sprawdza, że*
   - $b = 0$ $r \in \{0, \ldots, \beta\}$ *i że* $\mathcal{B}$ *zawierały* x^r *i* $x^{r-\beta}$
   - $b = 1$ $\widetilde{r} \in \Omega$ *oraz że jedno z* $\mathcal{B}$ *zawierało* $\widetilde{z}$ *takie, że* $x^{\widetilde{r}} \equiv \widetilde{z}y$

[a] $\mathcal{B}$ to "computationally secure blob"

Protokół potwierdzenia dla schematu (2)

Protokół (2)

Prover $\mathcal{P}$		*Verifier* $\mathcal{V}$
	$\xleftarrow{a \equiv S^r}$	*wybiera* $r \in \{1, \ldots, N\}$
$b :\equiv a^{v/s}$	$\xrightarrow{\mathcal{B}(b)}$	
sprawdza, że $a \equiv S^r$	$\xleftarrow{r}$	
	ujawnia b $\xrightarrow{}$	*sprawdza* b *i że* $b \equiv m^{vr}$

Obie części protokołu potwierdzenia podpisu są complete, sound i (computationally) zero-knowledge.

Protokół potwierdzenia dla schematu (2)

Protokół (2)

Prover $\mathcal{P}$		*Verifier* $\mathcal{V}$
	$a \equiv S^r \longleftarrow$	*wybiera* $r \in \{1, \ldots, N\}$
$b :\equiv a^{v/s}$	$\mathcal{B}(b) \longrightarrow$	
sprawdza, że $a \equiv S^r$	$\longleftarrow r$	
	ujawnia b $\longrightarrow$	*sprawdza b i że* $b \equiv m^{vr}$

Obie części protokołu potwierdzenia podpisu są complete, sound i (computationally) zero-knowledge.

Protokół potwierdzenia dla schematu (2)

Protokół (2)

Prover $\mathcal{P}$		*Verifier* $\mathcal{V}$
	$\xleftarrow{a \equiv S^r}$	*wybiera* $r \in \{1, \ldots, N\}$
$b :\equiv a^{v/s}$	$\xrightarrow{\mathcal{B}(b)}$	
sprawdza, że $a \equiv S^r$	$\xleftarrow{r}$	
	ujawnia b $\longrightarrow$	*sprawdza b i że* $b \equiv m^{vr}$

Obie części protokołu potwierdzenia podpisu są complete, sound i (computationally) zero-knowledge.

Protokół potwierdzenia dla schematu (2)

Protokół (2)

Prover $\mathcal{P}$		*Verifier* $\mathcal{V}$
	$\xleftarrow{a \equiv S^r}$	*wybiera* $r \in \{1, \ldots, N\}$
$b :\equiv a^{v/s}$	$\xrightarrow{\mathcal{B}(b)}$	
sprawdza, że $a \equiv S^r$	$\xleftarrow{r}$	
	ujawnia b $\longrightarrow$	*sprawdza* b *i że* $b \equiv m^{vr}$

Obie części protokołu potwierdzenia podpisu są complete, sound i (computationally) zero-knowledge.

Protokół potwierdzenia dla schematu (2)

Protokół (2)

Prover $\mathcal{P}$		*Verifier* $\mathcal{V}$
	$\xleftarrow{a \equiv S^r}$	*wybiera* $r \in \{1, \ldots, N\}$
$b :\equiv a^{v/s}$	$\xrightarrow{\mathcal{B}(b)}$	
sprawdza, że $a \equiv S^r$	$\xleftarrow{r}$	
	ujawnia b $\xrightarrow{}$	*sprawdza* b *i że* $b \equiv m^{vr}$

Obie części protokołu potwierdzenia podpisu są complete, sound i (computationally) zero-knowledge.

Protokół potwierdzenia dla schematu (2)

Protokół (2)

Prover $\mathcal{P}$		*Verifier* $\mathcal{V}$
	$\xleftarrow{a \equiv S^r}$	*wybiera* $r \in \{1, \ldots, N\}$
$b :\equiv a^{v/s}$	$\xrightarrow{\mathcal{B}(b)}$	
sprawdza, że $a \equiv S^r$	$\xleftarrow{r}$	
	ujawnia b $\xrightarrow{}$	*sprawdza* b *i że* $b \equiv m^{vr}$

Obie części protokołu potwierdzenia podpisu są complete, sound i (computationally) zero-knowledge.

Protokół wyparcia dla schematu (2)

Protokół (3)

$$\begin{array}{lll} \textit{sekret } \mathcal{P} & : & s \\ \textit{publiczne} & : & N, v, m, S, \Omega; \quad m, s \in \mathbb{Z}_N^* \\ \mathcal{P} \textit{ dowodzi } \mathcal{V} & : & S \neq m^s \mod N \wedge s \in \Phi \wedge s | v \end{array}$$

Nie ma dowodów z wiedzą zerową, że $\alpha^x \neq \beta^x \mod N$, przy danych $\{N, \alpha, \beta, \alpha^x\}$ i nieznanym $\phi(N)$.
Aby przeprowadzić dowód, TA publikuje jeszcze

- $(\widetilde{g}, \widetilde{h})$ - generujące $\mathbb{Z}_N^*$
- pełną listę elementów $(i, \widetilde{g}^{s_i}, \widetilde{h}^{s_i})$

Protokół wyparcia dla schematu (2)

Protokół (3)

sekret $\mathcal{P}$	:	s
publiczne	:	$N, v, m, S, \Omega; \quad m, s \in \mathbb{Z}_N^*$
$\mathcal{P}$ *dowodzi* $\mathcal{V}$	:	$S \neq m^s \mod N \wedge s \in \Phi \wedge s \mid v$

Nie ma dowodów z wiedzą zerową, że $\alpha^x \neq \beta^x \mod N$, przy danych $\{N, \alpha, \beta, \alpha^x\}$ i nieznanym $\phi(N)$.

Aby przeprowadzić dowód, TA publikuje jeszcze

- $(\widetilde{g}, \widetilde{h})$ - generujące $\mathbb{Z}_N^*$
- pełną listę elementów $(i, \widetilde{g}^{s_i}, \widetilde{h}^{s_i})$

Protokół wyparcia dla schematu (2)

Protokół (3)

sekret $\mathcal{P}$	:	s
publiczne	:	$N, v, m, S, \Omega; \quad m, s \in \mathbb{Z}_N^*$
$\mathcal{P}$ *dowodzi* $\mathcal{V}$	:	$S \neq m^s \mod N \wedge s \in \Phi \wedge s \mid v$

Nie ma dowodów z wiedzą zerową, że $\alpha^x \neq \beta^x \mod N$, przy danych $\{N, \alpha, \beta, \alpha^x\}$ i nieznanym $\phi(N)$.
Aby przeprowadzić dowód, TA publikuje jeszcze

- $(\widetilde{g}, \widetilde{h})$ - generujące $\mathbb{Z}_N^*$
- pełną listę elementów $(i, \widetilde{g}^{s_i}, \widetilde{h}^{s_i})$

Protokół wyparcia dla schematu (2)

Protokół (3)

sekret $\mathcal{P}$	:	s
publiczne	:	$N, v, m, S, \Omega; \quad m, s \in \mathbb{Z}_N^*$
$\mathcal{P}$ *dowodzi* $\mathcal{V}$	:	$S \neq m^s \mod N \wedge s \in \Phi \wedge s \mid v$

Nie ma dowodów z wiedzą zerową, że $\alpha^x \neq \beta^x \mod N$, przy danych $\{N, \alpha, \beta, \alpha^x\}$ i nieznanym $\phi(N)$.
Aby przeprowadzić dowód, TA publikuje jeszcze

- $(\widetilde{g}, \widetilde{h})$ - generujące $\mathbb{Z}_N^*$
- pełną listę elementów $(i, \widetilde{g}^{s_i}, \widetilde{h}^{s_i})$

Protokół wyparcia dla schematu (2)

Protokół (3)

sekret $\mathcal{P}$	:	s
publiczne	:	$N, v, m, S, \Omega; \quad m, s \in \mathbb{Z}_N^*$
$\mathcal{P}$ *dowodzi* $\mathcal{V}$	:	$S \neq m^s \mod N \wedge s \in \Phi \wedge s \mid v$

Nie ma dowodów z wiedzą zerową, że $\alpha^x \neq \beta^x \mod N$, przy danych $\{N, \alpha, \beta, \alpha^x\}$ i nieznanym $\phi(N)$.
Aby przeprowadzić dowód, TA publikuje jeszcze

- $(\widetilde{g}, \widetilde{h})$ - generujące $\mathbb{Z}_N^*$
- pełną listę elementów $(i, \widetilde{g}^{s_i}, \widetilde{h}^{s_i})$

Protokół wyparcia dla schematu (2)

Protokół (3)

Prover $\mathcal{P}$		Verifier $\mathcal{V}$
	$\longleftarrow$	
	$m^a \widetilde{g}^{r_1} \widetilde{h}^{r_2}$	wybiera[a] $a \in \{0, \ldots, l\}$
	$S^a (\widetilde{g}^s)^{r_1} (\widetilde{h}^s)^{r_2}$	oraz $r_1, r_2 \in \{1, \ldots, N\}$
oblicza a z $(\frac{m^s}{S})^a$		
	$\xrightarrow{\mathcal{B}(a)}$	
sprawdza r_1, r_2	$\xleftarrow{r_1, r_2}$	
	ujawnia a $\longrightarrow$	
		sprawdza a

[a] l jest dobrane tak, by sprawdzenie l wartości a było możliwe

Jeśli to $\mathcal{P}$ podpisał wiadomośc, musi zgadywać wartość a, bo $(\frac{m^s}{S})^a \equiv 1$.

Protokół wyparcia dla schematu (2)

Protokół (3)

Prover $\mathcal{P}$		*Verifier* $\mathcal{V}$
	$\longleftarrow$ $m^a \widetilde{g}^{r_1} \widetilde{h}^{r_2}$	*wybiera*[a] $a \in \{0, \ldots, l\}$
	$S^a (\widetilde{g}^s)^{r_1} (\widetilde{h}^s)^{r_2}$	*oraz* $r_1, r_2 \in \{1, \ldots, N\}$
oblicza a *z* $(\frac{m^s}{S})^a$		
	$\xrightarrow{\mathcal{B}(a)}$	
sprawdza r_1, r_2	$\xleftarrow{r_1, r_2}$	
	ujawnia a $\longrightarrow$	*sprawdza* a

[a] l jest dobrane tak, by sprawdzenie l wartości a było możliwe

Jeśli to $\mathcal{P}$ podpisał wiadomośc, musi zgadywać wartość a, bo $(\frac{m^s}{S})^a \equiv 1$.

Protokół wyparcia dla schematu (2)

Protokół (3)

Prover $\mathcal{P}$		*Verifier* $\mathcal{V}$
	$\longleftarrow$	
	$m^a \widetilde{g}^{r_1} \widetilde{h}^{r_2}$	*wybiera*[a] $a \in \{0, \ldots, l\}$
	$S^a (\widetilde{g}^s)^{r_1} (\widetilde{h}^s)^{r_2}$	*oraz* $r_1, r_2 \in \{1, \ldots, N\}$
oblicza a *z* $(\frac{m^s}{S})^a$		
	$\xrightarrow{\mathcal{B}(a)}$	
sprawdza r_1, r_2	$\xleftarrow{r_1, r_2}$	
	ujawnia a $\longrightarrow$	
		sprawdza a

[a] l jest dobrane tak, by sprawdzenie l wartości a było możliwe

Jeśli to $\mathcal{P}$ podpisał wiadomośc, musi zgadywać wartość a, bo $(\frac{m^s}{S})^a \equiv 1$.

Protokół wyparcia dla schematu (2)

Protokół (3)

Prover $\mathcal{P}$		*Verifier* $\mathcal{V}$
	$\longleftarrow$	
	$m^a \widetilde{g}^{r_1} \widetilde{h}^{r_2}$	*wybiera*[a] $a \in \{0, \ldots, l\}$
	$S^a (\widetilde{g}^s)^{r_1} (\widetilde{h}^s)^{r_2}$	*oraz* $r_1, r_2 \in \{1, \ldots, N\}$
oblicza a z $(\frac{m^s}{S})^a$		
	$\xrightarrow{\mathcal{B}(a)}$	
sprawdza r_1, r_2	$\xleftarrow{r_1, r_2}$	
	ujawnia a $\longrightarrow$	
		sprawdza a

[a] l jest dobrane tak, by sprawdzenie l wartości a było możliwe

Jeśli to $\mathcal{P}$ podpisał wiadomośc, musi zgadywać wartość a, bo $(\frac{m^s}{S})^a \equiv 1$.

Protokół wyparcia dla schematu (2)

Protokół (3)

Prover $\mathcal{P}$		*Verifier* $\mathcal{V}$
	$\longleftarrow$	
	$m^a \widetilde{g}^{r_1} \widetilde{h}^{r_2}$	*wybiera*[a] $a \in \{0, \ldots, l\}$
	$S^a (\widetilde{g}^s)^{r_1} (\widetilde{h}^s)^{r_2}$	*oraz* $r_1, r_2 \in \{1, \ldots, N\}$
oblicza a z $(\frac{m^s}{S})^a$		
	$\xrightarrow{\mathcal{B}(a)}$	
sprawdza r_1, r_2	$\xleftarrow{r_1, r_2}$	
	ujawnia a $\longrightarrow$	*sprawdza a*

[a] l jest dobrane tak, by sprawdzenie l wartości a było możliwe

Jeśli to $\mathcal{P}$ podpisał wiadomośc, musi zgadywać wartość a, bo $(\frac{m^s}{S})^a \equiv 1$.

Protokół wyparcia dla schematu (2)

Protokół (3)

Prover $\mathcal{P}$		*Verifier* $\mathcal{V}$
	$\longleftarrow$	
	$m^a \widetilde{g}^{r_1} \widetilde{h}^{r_2}$	*wybiera*[a] $a \in \{0, \ldots, l\}$
	$S^a (\widetilde{g}^s)^{r_1} (\widetilde{h}^s)^{r_2}$	*oraz* $r_1, r_2 \in \{1, \ldots, N\}$
oblicza a z $(\frac{m^s}{S})^a$		
	$\xrightarrow{\mathcal{B}(a)}$	
sprawdza r_1, r_2	$\xleftarrow{r_1, r_2}$	
	ujawnia a $\longrightarrow$	
		sprawdza a

[a] l jest dobrane tak, by sprawdzenie l wartości a było możliwe

Jeśli to $\mathcal{P}$ podpisał wiadomośc, musi zgadywać wartość a, bo $(\frac{m^s}{S})^a \equiv 1$.

Protokół wyparcia dla schematu (2)

Protokół (3)

Prover $\mathcal{P}$		*Verifier* $\mathcal{V}$
	$\longleftarrow$	
	$m^a \widetilde{g}^{r_1} \widetilde{h}^{r_2}$	*wybiera*[a] $a \in \{0, \ldots, l\}$
	$S^a (\widetilde{g}^s)^{r_1} (\widetilde{h}^s)^{r_2}$	*oraz* $r_1, r_2 \in \{1, \ldots, N\}$
oblicza a z $(\frac{m^s}{S})^a$		
	$\xrightarrow{\mathcal{B}(a)}$	
sprawdza r_1, r_2	$\xleftarrow{r_1, r_2}$	
	ujawnia a $\longrightarrow$	
		sprawdza a

[a] l jest dobrane tak, by sprawdzenie l wartości a było możliwe

Jeśli to $\mathcal{P}$ podpisał wiadomośc, musi zgadywać wartość a, bo $(\frac{m^s}{S})^a = 1$.

Uwagi nt. schematu (2)

- Gdy wszyscy członkowie grupy poza i są w zmowie, mogą poznać klucz i. Ale TA może być członkiem grupy: $v = s_{TA} \cdot \prod s_i$. Pozwala to na stosowanie grup dwuosobowych.
- $|v|$ zależy liniowo od liczby osób w grupie, więc ilość obliczeń w protokole potwierdzenia podpisu jest liniowa wzgl. tej liczby.

Uwagi nt. schematu (2)

- Gdy wszyscy członkowie grupy poza *i* są w zmowie, mogą poznać klucz *i*. Ale TA może być członkiem grupy: $v = s_{TA} \cdot \prod s_i$. Pozwala to na stosowanie grup dwuosobowych.
- $|v|$ zależy liniowo od liczby osób w grupie, więc ilość obliczeń w protokole potwierdzenia podpisu jest liniowa wzgl. tej liczby.

Uwagi nt. schematu (2)

- Gdy wszyscy członkowie grupy poza i są w zmowie, mogą poznać klucz i. Ale TA może być członkiem grupy: $v = s_{TA} \cdot \prod s_i$. Pozwala to na stosowanie grup dwuosobowych.
- $|v|$ zależy liniowo od liczby osób w grupie, więc ilość obliczeń w protokole potwierdzenia podpisu jest liniowa wzgl. tej liczby.

Uwagi nt. schematu (2)

- Gdy wszyscy członkowie grupy poza i są w zmowie, mogą poznać klucz i. Ale TA może być członkiem grupy: $v = s_{TA} \cdot \prod s_i$. Pozwala to na stosowanie grup dwuosobowych.
- $|v|$ zależy liniowo od liczby osób w grupie, więc ilość obliczeń w protokole potwierdzenia podpisu jest liniowa wzgl. tej liczby.

Uwagi nt. schematu (2), cd.

- Zbiór Φ może być inny, ale musi spełniać pewne warunki. Niech $\Phi = \{\phi_1, \ldots, \phi_1 + \phi_2\}$. Wtedy $1, N, \phi_1^2 \notin \widetilde{\Phi} = \{\phi_1 - \phi_2, \ldots, \phi_1 + 2\phi_2\}$. Uniemożliwia to
  - 1 dowodzenie znajomości $s = 1$
  - ϕ_1^2 współpracę członków i oraz j, którzy mogą stworzyć podpis $S \equiv m^{s_i s_j}$, którego obaj mogą się wyprzeć
- TA musi zadbać o odpowiednią postać generatorów, by nie zdradzić faktoryzacji N.

Uwagi nt. schematu (2), cd.

- Zbiór Φ może być inny, ale musi spełniać pewne warunki. Niech $\Phi = \{\phi_1, \ldots, \phi_1 + \phi_2\}$. Wtedy $1, N, \phi_1^2 \notin \widetilde{\Phi} = \{\phi_1 - \phi_2, \ldots, \phi_1 + 2\phi_2\}$. Uniemożliwia to
  - 1 dowodzenie znajomości $s = 1$
  - ϕ_1^2 współpracę członków i oraz j, którzy mogą stworzyć podpis $S \equiv m^{s_i s_j}$, którego obaj mogą się wyprzeć
- TA musi zadbać o odpowiednią postać generatorów, by nie zdradzić faktoryzacji N.

Uwagi nt. schematu (2), cd.

- Zbiór Φ może być inny, ale musi spełniać pewne warunki. Niech $\Phi = \{\phi_1, \ldots, \phi_1 + \phi_2\}$. Wtedy $1, N, \phi_1^2 \notin \widetilde{\Phi} = \{\phi_1 - \phi_2, \ldots, \phi_1 + 2\phi_2\}$. Uniemożliwia to
  - 1 dowodzenie znajomości $s = 1$
  - ϕ_1^2 współpracę członków i oraz j, którzy mogą stworzyć podpis $S \equiv m^{s_i s_j}$, którego obaj mogą się wyprzeć
- TA musi zadbać o odpowiednią postać generatorów, by nie zdradzić faktoryzacji N.

Uwagi nt. schematu (2), cd.

- Zbiór Φ może być inny, ale musi spełniać pewne warunki. Niech $\Phi = \{\phi_1, \ldots, \phi_1 + \phi_2\}$. Wtedy $1, N, \phi_1^2 \notin \widetilde{\Phi} = \{\phi_1 - \phi_2, \ldots, \phi_1 + 2\phi_2\}$. Uniemożliwia to
  - 1 dowodzenie znajomości $s = 1$
  - ϕ_1^2 współpracę członków i oraz j, którzy mogą stworzyć podpis $S \equiv m^{s_i s_j}$, którego obaj mogą się wyprzeć
- TA musi zadbać o odpowiednią postać generatorów, by nie zdradzić faktoryzacji N.

Uwagi nt. schematu (2), cd.

- Zbiór Φ może być inny, ale musi spełniać pewne warunki. Niech $\Phi = \{\phi_1, \ldots, \phi_1 + \phi_2\}$. Wtedy $1, N, \phi_1^2 \notin \widetilde{\Phi} = \{\phi_1 - \phi_2, \ldots, \phi_1 + 2\phi_2\}$. Uniemożliwia to
  - 1 dowodzenie znajomości $s = 1$
  - ϕ_1^2 współpracę członków i oraz j, którzy mogą stworzyć podpis $S \equiv m^{s_i s_j}$, którego obaj mogą się wyprzeć
- TA musi zadbać o odpowiednią postać generatorów, by nie zdradzić faktoryzacji N.

Schemat (3)

1. Każdy członek grupy, i, wybiera swoje $N_i = p_i q_i$.
   - kluczem publicznym i jest N_i
   - kluczem prywatnym i jest p_i.
   - p_i oraz q_i muszą spełniać dodatkowe warunki
     - $\forall_i p_i \in \Phi = \{\lceil\sqrt{M}\rceil, \ldots, \lfloor 2\sqrt{M}\rfloor - 1\}$
     - $\forall_i q_i > 4\sqrt{M}$
2. TA wybiera $N = pq$ i f.
3. Podpisem członka i pod wiadomością n będzie $\Gamma, (f(n))^{s_i} \bmod N$, gdzie Γ jest losowanym przez niego podzbiorem członków grupy.

Schemat (3)

1. Każdy członek grupy, i, wybiera swoje $N_i = p_i q_i$.
   - kluczem publicznym i jest N_i
   - kluczem prywatnym i jest p_i.
   - p_i oraz q_i muszą spełniać dodatkowe warunki
     - $\forall_i p_i \in \Phi = \{\lceil\sqrt{M}\rceil, \ldots, \lfloor 2\sqrt{M}\rfloor - 1\}$
     - $\forall_i q_i > 4\sqrt{M}$
2. TA wybiera $N = pq$ i f.
3. Podpisem członka i pod wiadomością n będzie $\Gamma, (f(n))^{s_i} \bmod N$, gdzie Γ jest losowanym przez niego podzbiorem członków grupy.

Schemat (3)

1. Każdy członek grupy, i, wybiera swoje $N_i = p_i q_i$.
   - kluczem publicznym i jest N_i
   - kluczem prywatnym i jest p_i.
   - p_i oraz q_i muszą spełniać dodatkowe warunki
     - $\forall_i p_i \in \Phi = \{\lceil\sqrt{M}\rceil, \ldots, \lfloor 2\sqrt{M}\rfloor - 1\}$
     - $\forall_i q_i > 4\sqrt{M}$
2. TA wybiera $N = pq$ i f.
3. Podpisem członka i pod wiadomością n będzie $\Gamma, (f(n))^{s_i} \bmod N$, gdzie Γ jest losowanym przez niego podzbiorem członków grupy.

Schemat (3)

1. Każdy członek grupy, i, wybiera swoje $N_i = p_i q_i$.
   - kluczem publicznym i jest N_i
   - kluczem prywatnym i jest p_i.
   - p_i oraz q_i muszą spełniać dodatkowe warunki
     - $\forall_i p_i \in \Phi = \{\lceil\sqrt{M}\rceil, \ldots, \lfloor 2\sqrt{M}\rfloor - 1\}$
     - $\forall_i q_i > 4\sqrt{M}$
2. TA wybiera $N = pq$ i f.
3. Podpisem członka i pod wiadomością n będzie $\Gamma, (f(n))^{s_i} \bmod N$, gdzie Γ jest losowanym przez niego podzbiorem członków grupy.

Schemat (3)

1. Każdy członek grupy, i, wybiera swoje $N_i = p_i q_i$.
   - kluczem publicznym i jest N_i
   - kluczem prywatnym i jest p_i.
   - p_i oraz q_i muszą spełniać dodatkowe warunki
     - $\forall_i p_i \in \Phi = \{\lceil\sqrt{M}\rceil, \ldots, \lfloor 2\sqrt{M}\rfloor - 1\}$
     - $\forall_i q_i > 4\sqrt{M}$
2. TA wybiera $N = pq$ i f.
3. Podpisem członka i pod wiadomością n będzie $\Gamma, (f(n))^{s_i} \bmod N$, gdzie Γ jest losowanym przez niego podzbiorem członków grupy.

Schemat (3)

1. Każdy członek grupy, i, wybiera swoje $N_i = p_i q_i$.
   - kluczem publicznym i jest N_i
   - kluczem prywatnym i jest p_i.
   - p_i oraz q_i muszą spełniać dodatkowe warunki
     - $\forall_i p_i \in \Phi = \{\lceil\sqrt{M}\rceil, \ldots, \lfloor 2\sqrt{M}\rfloor - 1\}$
     - $\forall_i q_i > 4\sqrt{M}$
2. TA wybiera $N = pq$ i f.
3. Podpisem członka i pod wiadomością n będzie $\Gamma, (f(n))^{s_i} \bmod N$, gdzie Γ jest losowanym przez niego podzbiorem członków grupy.

Schemat (3)

1. Każdy członek grupy, i, wybiera swoje $N_i = p_i q_i$.
   - kluczem publicznym i jest N_i
   - kluczem prywatnym i jest p_i.
   - p_i oraz q_i muszą spełniać dodatkowe warunki
     - $\forall_i p_i \in \Phi = \{\lceil\sqrt{M}\rceil, \ldots, \lfloor 2\sqrt{M}\rfloor - 1\}$
     - $\forall_i q_i > 4\sqrt{M}$
2. TA wybiera $N = pq$ i f.
3. Podpisem członka i pod wiadomością n będzie $\Gamma, (f(n))^{s_i} \bmod N$, gdzie Γ jest losowanym przez niego podzbiorem członków grupy.

Schemat (3)

1. Każdy członek grupy, i, wybiera swoje $N_i = p_i q_i$.
   - kluczem publicznym i jest N_i
   - kluczem prywatnym i jest p_i.
   - p_i oraz q_i muszą spełniać dodatkowe warunki
     - $\forall_i p_i \in \Phi = \{\lceil\sqrt{M}\rceil, \ldots, \lfloor 2\sqrt{M}\rfloor - 1\}$
     - $\forall_i q_i > 4\sqrt{M}$
2. TA wybiera $N = pq$ i f.
3. Podpisem członka i pod wiadomością n będzie $\Gamma, (f(n))^{s_i} \bmod N$, gdzie Γ jest losowanym przez niego podzbiorem członków grupy.

Schemat (3)

1. Każdy członek grupy, i, wybiera swoje $N_i = p_i q_i$.
   - kluczem publicznym i jest N_i
   - kluczem prywatnym i jest p_i.
   - p_i oraz q_i muszą spełniać dodatkowe warunki
     - $\forall_i p_i \in \Phi = \{\lceil\sqrt{M}\rceil, \ldots, \lfloor 2\sqrt{M}\rfloor - 1\}$
     - $\forall_i q_i > 4\sqrt{M}$
2. TA wybiera $N = pq$ i f.
3. Podpisem członka i pod wiadomością n będzie $\Gamma, (f(n))^{s_i} \bmod N$, gdzie Γ jest losowanym przez niego podzbiorem członków grupy.

Czego dowodzi osoba podpisująca

Osoba podpisująca wiadomość dowodzi znajomości s takiego, że

- $S \equiv m^s \mod N$
- $s \in \Phi$
- $s|v := \prod_{j \in \Gamma} N_j$

Gdzie publicznie znane są: N, v, $m = f(N)$, S, Φ; $m, S \in \mathbb{Z}_N^*$.
Z zakładanych nierówności wynika, że jedyne $s \in \Phi$ jakie może znać i, to p_i, bo $\Phi = \{\lceil\sqrt{M}\rceil, \ldots, \lfloor 3\sqrt{M}\rfloor\}$.

Czego dowodzi osoba podpisująca

Osoba podpisująca wiadomość dowodzi znajomości s takiego, że

- $S \equiv m^s \mod N$
- $s \in \Phi$
- $s|v := \prod_{j \in \Gamma} N_j$

Gdzie publicznie znane są: N, v, $m = f(N)$, S, Φ; $m, S \in \mathbb{Z}_N^*$.
Z zakładanych nierówności wynika, że jedyne $s \in \Phi$ jakie może znać i, to p_i, bo $\Phi = \{\lceil\sqrt{M}\rceil, \ldots, \lfloor 3\sqrt{M}\rfloor\}$.

Czego dowodzi osoba podpisująca

Osoba podpisująca wiadomość dowodzi znajomości s takiego, że

- $S \equiv m^s \mod N$
- $s \in \Phi$
- $s|v := \prod_{j \in \Gamma} N_j$

Gdzie publicznie znane są: N, v, $m = f(N)$, S, Φ; $m, S \in \mathbb{Z}_N^*$.
Z zakładanych nierówności wynika, że jedyne $s \in \Phi$ jakie może znać i, to p_i, bo $\Phi = \{\lceil\sqrt{M}\rceil, \ldots, \lfloor 3\sqrt{M}\rfloor\}$.

Czego dowodzi osoba podpisująca

Osoba podpisująca wiadomość dowodzi znajomości s takiego, że

- $S \equiv m^s \mod N$
- $s \in \Phi$
- $s|v := \prod_{j \in \Gamma} N_j$

Gdzie publicznie znane są: N, v, $m = f(N)$, S, Φ; $m, S \in \mathbb{Z}_N^*$.
Z zakładanych nierówności wynika, że jedyne $s \in \Phi$ jakie może znać i, to p_i, bo $\Phi = \{\lceil\sqrt{M}\rceil, \ldots, \lfloor 3\sqrt{M}\rfloor\}$.

Czego dowodzi osoba podpisująca

Osoba podpisująca wiadomość dowodzi znajomości s takiego, że

- $S \equiv m^s \mod N$
- $s \in \Phi$
- $s|v := \prod_{j \in \Gamma} N_j$

Gdzie publicznie znane są: N, v, $m = f(N)$, S, Φ; $m, S \in \mathbb{Z}_N^*$.
Z zakładanych nierówności wynika, że jedyne $s \in \Phi$ jakie może znać i, to p_i, bo $\Phi = \{\lceil\sqrt{M}\rceil, \ldots, \lfloor 3\sqrt{M}\rfloor\}$.

Czego dowodzi osoba podpisująca

Osoba podpisująca wiadomość dowodzi znajomości s takiego, że

- $S \equiv m^s \mod N$
- $s \in \Phi$
- $s|v := \prod_{j \in \Gamma} N_j$

Gdzie publicznie znane są: N, v, $m = f(N)$, S, Φ; $m, S \in \mathbb{Z}_N^*$.
Z zakładanych nierówności wynika, że jedyne $s \in \Phi$ jakie może znać i, to p_i, bo $\Phi = \{\lceil\sqrt{M}\rceil, \ldots, \lfloor 3\sqrt{M}\rfloor\}$.

Szkic schematu (4)

- p jest pierwsze, g i h są publicznym generatorami $\mathbb{Z}_p^*$
- kluczem prywatnym i jest s_i, publicznym $k_i \equiv g^{s_i} \mod p$
- podpisem pod n jest $\Gamma, f(n)_i^s \mod p$
- $\mathcal{P}$ dowodzi $\mathcal{V}$ znajomości s takiego, że
  - $S \equiv m^s \mod p$
  - $g^s \in \{k_j : j \in \Gamma\}$

Szkic schematu (4)

- p jest pierwsze, g i h są publicznym generatorami $\mathbb{Z}_p^*$
- kluczem prywatnym i jest s_i, publicznym $k_i \equiv g^{s_i} \mod p$
- podpisem pod n jest $\Gamma, f(n)_i^s \mod p$
- $\mathcal{P}$ dowodzi $\mathcal{V}$ znajomości s takiego, że
  - $S \equiv m^s \mod p$
  - $g^s \in \{k_j : j \in \Gamma\}$

Porównanie pierwszych schematów

nr sch.	zał. dot. bezp.	TA potrz. do otw.	grupa stała	typ podp.	$\lvert\mathcal{K}\rvert$	# przesł. bitów	# obl.
(1)	dow	tak	tak	dow	lin	const	const
(2)	1	nie	tak	niezap	lin	lin	const
(3)	1	nie	nie	niezap	lin	lin	const
(4)	2	nie	nie	niezap	lin	lin	lin

Podwójne podpisy

- Niech prywatnym kluczem P_i będzie s_i, publicznym grupy – K a podpis pod wiadomością m, $\sigma_K(m)$, spełnia własności z definicji, ale nie umożliwia identyfikacji osoby, która złożyła podpis.
- Rozważmy taki schemat: P_i ma prywatne klucze s_i, t_i, grupa – K_1, K_2; P_i podpisuje wiadomość oboma kluczami, zaś TA zna jego klucz prywatny t_i.
- TA jest w stanie zidentyfikować osobę, która złożyła podpis pod m nie kontaktując się z nikim a jednocześnie nie umie podrabiać podpisów.
- Co więcej, klucz t_i można podzielić między członków grupy tak, by każdych k z nich było w stanie odtworzyć t_i. Więc TA wcale nie jest potrzebne!

Podwójne podpisy

- Niech prywatnym kluczem P_i będzie s_i, publicznym grupy – K a podpis pod wiadomością m, $\sigma_K(m)$, spełnia własności z definicji, ale nie umożliwia identyfikacji osoby, która złożyła podpis.
- Rozważmy taki schemat: P_i ma prywatne klucze s_i, t_i, grupa – K_1, K_2; P_i podpisuje wiadomość oboma kluczami, zaś TA zna jego klucz prywatny t_i.
- TA jest w stanie zidentyfikować osobę, która złożyła podpis pod m nie kontaktując się z nikim a jednocześnie nie umie podrabiać podpisów.
- Co więcej, klucz t_i można podzielić między członków grupy tak, by każdych k z nich było w stanie odtworzyć t_i. Więc TA wcale nie jest potrzebne!

Podwójne podpisy

- Niech prywatnym kluczem P_i będzie s_i, publicznym grupy – K a podpis pod wiadomością m, $\sigma_K(m)$, spełnia własności z definicji, ale nie umożliwia identyfikacji osoby, która złożyła podpis.
- Rozważmy taki schemat: P_i ma prywatne klucze s_i, t_i, grupa – K_1, K_2; P_i podpisuje wiadomość oboma kluczami, zaś TA zna jego klucz prywatny t_i.
- TA jest w stanie zidentyfikować osobę, która złożyła podpis pod m nie kontaktując się z nikim a jednocześnie nie umie podrabiać podpisów.
- Co więcej, klucz t_i można podzielić między członków grupy tak, by każdych k z nich było w stanie odtworzyć t_i. Więc TA wcale nie jest potrzebne!

Podwójne podpisy

- Niech prywatnym kluczem P_i będzie s_i, publicznym grupy – K a podpis pod wiadomością m, $\sigma_K(m)$, spełnia własności z definicji, ale nie umożliwia identyfikacji osoby, która złożyła podpis.
- Rozważmy taki schemat: P_i ma prywatne klucze s_i, t_i, grupa – K_1, K_2; P_i podpisuje wiadomość oboma kluczami, zaś TA zna jego klucz prywatny t_i.
- TA jest w stanie zidentyfikować osobę, która złożyła podpis pod m nie kontaktując się z nikim a jednocześnie nie umie podrabiać podpisów.
- Co więcej, klucz t_i można podzielić między członków grupy tak, by każdych k z nich było w stanie odtworzyć t_i. Więc TA wcale nie jest potrzebne!

Podwójne podpisy

- Niech prywatnym kluczem P_i będzie s_i, publicznym grupy – K a podpis pod wiadomością m, $\sigma_K(m)$, spełnia własności z definicji, ale nie umożliwia identyfikacji osoby, która złożyła podpis.
- Rozważmy taki schemat: P_i ma prywatne klucze s_i, t_i, grupa – K_1, K_2; P_i podpisuje wiadomość oboma kluczami, zaś TA zna jego klucz prywatny t_i.
- TA jest w stanie zidentyfikować osobę, która złożyła podpis pod m nie kontaktując się z nikim a jednocześnie nie umie podrabiać podpisów.
- Co więcej, klucz t_i można podzielić między członków grupy tak, by każdych k z nich było w stanie odtworzyć t_i. Więc TA wcale nie jest potrzebne!

Podwójne podpisy

- Niech prywatnym kluczem P_i będzie s_i, publicznym grupy – K a podpis pod wiadomością m, $\sigma_K(m)$, spełnia własności z definicji, ale nie umożliwia identyfikacji osoby, która złożyła podpis.
- Rozważmy taki schemat: P_i ma prywatne klucze s_i, t_i, grupa – K_1, K_2; P_i podpisuje wiadomość oboma kluczami, zaś TA zna jego klucz prywatny t_i.
- TA jest w stanie zidentyfikować osobę, która złożyła podpis pod m nie kontaktując się z nikim a jednocześnie nie umie podrabiać podpisów.
- Co więcej, klucz t_i można podzielić między członków grupy tak, by każdych k z nich było w stanie odtworzyć t_i. Więc TA wcale nie jest potrzebne!

Heurystyka Fiata-Shamira

protokół identyfikacji $\longrightarrow$ schemat podpisów:

- $c := \mathcal{H}(x||m)$; m – wiadomość, x – świadectwo
- podpis pod m: (x, r); r – odpowiedź $\mathcal{P}$ na c
- funkcja $\mathcal{H}$ „pełni rolę weryfikatora”.
- To samo rozwiązanie można zastosować do podpisów grupowych!

Heurystyka Fiata-Shamira

protokół identyfikacji $\longrightarrow$ schemat podpisów:

- $c := \mathcal{H}(x||m)$; m – wiadomość, x – świadectwo
- podpis pod m: (x, r); r – odpowiedź $\mathcal{P}$ na c
- funkcja $\mathcal{H}$ „pełni rolę weryfikatora”.
- To samo rozwiązanie można zastosować do podpisów grupowych!

Heurystyka Fiata-Shamira

protokół identyfikacji $\longrightarrow$ schemat podpisów:

- $c := \mathcal{H}(x||m)$; m – wiadomość, x – świadectwo
- podpis pod m: (x, r); r – odpowiedź $\mathcal{P}$ na c
- funkcja $\mathcal{H}$ „pełni rolę weryfikatora”.
- To samo rozwiązanie można zastosować do podpisów grupowych!

Protokół Schoenmakersa

Protokół

1. *P losuje* $s_i, d_j \in \mathbb{Z}_q^*$ *dla* $i = 1, \ldots, n$ *oraz* $i = 2, \ldots, n$; *oblicza i wysyła do V:*
   - $a_1 = g^{s_1}$
   - $a_i = g^{s_i} h_i^{-d_i}$ *dla* $i = 2, 3, \ldots, n$
2. *V losuje* $c \in \mathbb{Z}_q^*$ *i wysyła do P*
3. *P oblicza* $d_1 = c - \sum_{i=2}^{n} d_i$ *oraz*
   - $r_1 = s_1 + x_1 d_1$
   - $r_i = s_i$ *dla* $i = 2, 3, \ldots, n$

   i wysyła $(d_1, \ldots, d_n, r_1, \ldots, r_n)$ *do V*
4. *V sprawdza, że*
   - $\sum_{i=1}^{n} = c$
   - $g^{r_i} = a_i h_i^{d_i}$ *dla* $i = 1, 2, \ldots, n$

Protokół Schoenmakersa

Protokół

1. *P losuje* $s_i, d_j \in \mathbb{Z}_q^*$ *dla* $i = 1, \ldots, n$ *oraz* $i = 2, \ldots, n$; *oblicza i wysyła do V:*
   - $a_1 = g^{s_1}$
   - $a_i = g^{s_i} h_i^{-d_i}$ *dla* $i = 2, 3, \ldots, n$
2. *V losuje* $c \in \mathbb{Z}_q^*$ *i wysyła do P*
3. *P oblicza* $d_1 = c - \sum_{i=2}^{n} d_i$ *oraz*
   - $r_1 = s_1 + x_1 d_1$
   - $r_i = s_i$ *dla* $i = 2, 3, \ldots, n$

   i wysyła $(d_1, \ldots, d_n, r_1, \ldots, r_n)$ *do V*
4. *V sprawdza, że*
   - $\sum_{i=1}^{n} = c$
   - $g^{r_i} = a_i h_i^{d_i}$ *dla* $i = 1, 2, \ldots, n$

Protokół Schoenmakersa

Protokół

1. *P losuje* $s_i, d_j \in \mathbb{Z}_q^*$ *dla* $i = 1, \ldots, n$ *oraz* $i = 2, \ldots, n$; *oblicza i wysyła do V:*
   - $a_1 = g^{s_1}$
   - $a_i = g^{s_i} h_i^{-d_i}$ *dla* $i = 2, 3, \ldots, n$
2. *V losuje* $c \in \mathbb{Z}_q^*$ *i wysyła do P*
3. *P oblicza* $d_1 = c - \sum_{i=2}^{n} d_i$ *oraz*
   - $r_1 = s_1 + x_1 d_1$
   - $r_i = s_i$ *dla* $i = 2, 3, \ldots, n$

   i wysyła $(d_1, \ldots, d_n, r_1, \ldots, r_n)$ *do V*
4. *V sprawdza, że*
   - $\sum_{i=1}^{n} = c$
   - $g^{r_i} = a_i h_i^{d_i}$ *dla* $i = 1, 2, \ldots, n$

Protokół Schoenmakersa

Protokół

1. *P losuje* $s_i, d_j \in \mathbb{Z}_q^*$ *dla* $i = 1, \ldots, n$ *oraz* $i = 2, \ldots, n$; *oblicza i wysyła do V:*
   - $a_1 = g^{s_1}$
   - $a_i = g^{s_i} h_i^{-d_i}$ *dla* $i = 2, 3, \ldots, n$
2. *V losuje* $c \in \mathbb{Z}_q^*$ *i wysyła do P*
3. *P oblicza* $d_1 = c - \sum_{i=2}^{n} d_i$ *oraz*
   - $r_1 = s_1 + x_1 d_1$
   - $r_i = s_i$ *dla* $i = 2, 3, \ldots, n$

   i wysyła $(d_1, \ldots, d_n, r_1, \ldots, r_n)$ *do V*
4. *V sprawdza, że*
   - $\sum_{i=1}^{n} = c$
   - $g^{r_i} = a_i h_i^{d_i}$ *dla* $i = 1, 2, \ldots, n$

Protokół Schoenmakersa

Protokół

1. *P losuje* $s_i, d_j \in \mathbb{Z}_q^*$ *dla* $i = 1, \ldots, n$ *oraz* $i = 2, \ldots, n$; *oblicza i wysyła do V:*
   - $a_1 = g^{s_1}$
   - $a_i = g^{s_i} h_i^{-d_i}$ *dla* $i = 2, 3, \ldots, n$
2. *V losuje* $c \in \mathbb{Z}_q^*$ *i wysyła do P*
3. *P oblicza* $d_1 = c - \sum_{i=2}^{n} d_i$ *oraz*
   - $r_1 = s_1 + x_1 d_1$
   - $r_i = s_i$ *dla* $i = 2, 3, \ldots, n$

   i wysyła $(d_1, \ldots, d_n, r_1, \ldots, r_n)$ *do V*
4. *V sprawdza, że*
   - $\sum_{i=1}^{n} = c$
   - $g^{r_i} = a_i h_i^{d_i}$ *dla* $i = 1, 2, \ldots, n$

Protokół Schoenmakersa

Protokół

1. *P losuje* $s_i, d_j \in \mathbb{Z}_q^*$ *dla* $i = 1, \ldots, n$ *oraz* $i = 2, \ldots, n$; *oblicza i wysyła do V:*
   - $a_1 = g^{s_1}$
   - $a_i = g^{s_i} h_i^{-d_i}$ *dla* $i = 2, 3, \ldots, n$
2. *V losuje* $c \in \mathbb{Z}_q^*$ *i wysyła do P*
3. *P oblicza* $d_1 = c - \sum_{i=2}^{n} d_i$ *oraz*
   - $r_1 = s_1 + x_1 d_1$
   - $r_i = s_i$ *dla* $i = 2, 3, \ldots, n$

   i wysyła $(d_1, \ldots, d_n, r_1, \ldots, r_n)$ *do V*
4. *V sprawdza, że*
   - $\sum_{i=1}^{n} = c$
   - $g^{r_i} = a_i h_i^{d_i}$ *dla* $i = 1, 2, \ldots, n$

Wstęp

- Dla przejrzystości zajmiemy się grupą dwuosobową.
- TA publikuje g_1 i g_2 – generatory G rzędu q
- kluczem prywatnym P_i jest (x_{i1}, x_{i2})
- kluczem publicznym P_i jest $h_i = g_1^{x_{i1}} g_2^{x_{i2}}$
- kluczem publicznym grupy jest (h_1, h_2)
- podpisem P_i pod $m = (m_1, m_2)$ jest $z = m_1^{x_{i1}} m_2^{x_{i2}}$
- P dowodzi znajomości x, y takich, że $z = m_1^x m_2^y$ i $g_1^x g_2^y = h_i$ dla $i = 1$ lub $i = 2$

Wstęp

- Dla przejrzystości zajmiemy się grupą dwuosobową.
- TA publikuje g_1 i g_2 – generatory G rzędu q
- kluczem prywatnym P_i jest (x_{i1}, x_{i2})
- kluczem publicznym P_i jest $h_i = g_1^{x_{i1}} g_2^{x_{i2}}$
- kluczem publicznym grupy jest (h_1, h_2)
- podpisem P_i pod $m = (m_1, m_2)$ jest $z = m_1^{x_{i1}} m_2^{x_{i2}}$
- P dowodzi znajomości x, y takich, że $z = m_1^x m_2^y$ i $g_1^x g_2^y = h_i$ dla $i = 1$ lub $i = 2$

Wstęp

- Dla przejrzystości zajmiemy się grupą dwuosobową.
- TA publikuje g_1 i g_2 – generatory G rzędu q
- kluczem prywatnym P_i jest (x_{i1}, x_{i2})
- kluczem publicznym P_i jest $h_i = g_1^{x_{i1}} g_2^{x_{i2}}$
- kluczem publicznym grupy jest (h_1, h_2)
- podpisem P_i pod $m = (m_1, m_2)$ jest $z = m_1^{x_{i1}} m_2^{x_{i2}}$
- P dowodzi znajomości x, y takich, że $z = m_1^x m_2^y$ i $g_1^x g_2^y = h_i$ dla $i = 1$ lub $i = 2$

Wstęp

- Dla przejrzystości zajmiemy się grupą dwuosobową.
- TA publikuje g_1 i g_2 – generatory G rzędu q
- kluczem prywatnym P_i jest (x_{i1}, x_{i2})
- kluczem publicznym P_i jest $h_i = g_1^{x_{i1}} g_2^{x_{i2}}$
- kluczem publicznym grupy jest (h_1, h_2)
- podpisem P_i pod $m = (m_1, m_2)$ jest $z = m_1^{x_{i1}} m_2^{x_{i2}}$
- P dowodzi znajomości x, y takich, że $z = m_1^x m_2^y$ i $g_1^x g_2^y = h_i$ dla $i = 1$ lub $i = 2$

Wstęp

- Dla przejrzystości zajmiemy się grupą dwuosobową.
- TA publikuje g_1 i g_2 – generatory G rzędu q
- kluczem prywatnym P_i jest (x_{i1}, x_{i2})
- kluczem publicznym P_i jest $h_i = g_1^{x_{i1}} g_2^{x_{i2}}$
- kluczem publicznym grupy jest (h_1, h_2)
- podpisem P_i pod $m = (m_1, m_2)$ jest $z = m_1^{x_{i1}} m_2^{x_{i2}}$
- P dowodzi znajomości x, y takich, że $z = m_1^x m_2^y$ i $g_1^x g_2^y = h_i$ dla $i = 1$ lub $i = 2$

Wstęp

- Dla przejrzystości zajmiemy się grupą dwuosobową.
- TA publikuje g_1 i g_2 – generatory G rzędu q
- kluczem prywatnym P_i jest (x_{i1}, x_{i2})
- kluczem publicznym P_i jest $h_i = g_1^{x_{i1}} g_2^{x_{i2}}$
- kluczem publicznym grupy jest (h_1, h_2)
- podpisem P_i pod $m = (m_1, m_2)$ jest $z = m_1^{x_{i1}} m_2^{x_{i2}}$
- P dowodzi znajomości x, y takich, że $z = m_1^x m_2^y$ i $g_1^x g_2^y = h_i$ dla $i = 1$ lub $i = 2$

Wstęp

- Dla przejrzystości zajmiemy się grupą dwuosobową.
- TA publikuje g_1 i g_2 – generatory G rzędu q
- kluczem prywatnym P_i jest (x_{i1}, x_{i2})
- kluczem publicznym P_i jest $h_i = g_1^{x_{i1}} g_2^{x_{i2}}$
- kluczem publicznym grupy jest (h_1, h_2)
- podpisem P_i pod $m = (m_1, m_2)$ jest $z = m_1^{x_{i1}} m_2^{x_{i2}}$
- P dowodzi znajomości x, y takich, że $z = m_1^x m_2^y$ i $g_1^x g_2^y = h_i$ dla $i = 1$ lub $i = 2$

Protokół

Protokół

1. P: $s_1, s_2, t_1, t_2, d_2 \in_{\mathcal{R}} \mathbb{Z}_q^*$
2. P liczy $a_1 := g_1^{s_1} g_2^{s_2}$, $b_1 := m_1^{s_1} m_2^{s_2}$
   oraz $a_2 := g_1^{t_1} g_2^{t_2} h_2^{-d_2}$, $b_2 := m_1^{t_1} m_2^{t_2} h_2^{-d_2}$
3. P wysyła do V: (a_1, a_2, b_1, b_2)
4. V wysyła do P: $c \in_{\mathcal{R}} \mathbb{Z}_q^*$
5. P liczy $d_1 := c - d_2$, $(u_1, u_2) := (t_1, t_2)$
   oraz $(r_1, r_2) := (s_1 + d_1 x_{11}, s_2 + d_1 x_{12})$
6. P wysyła do V: $(d_1, d_2, r_1, r_2, u_1, u_2)$

Protokół

Protokół

1. P: $s_1, s_2, t_1, t_2, d_2 \in_{\mathcal{R}} \mathbb{Z}_q^*$
2. P liczy $a_1 := g_1^{s_1} g_2^{s_2}$, $b_1 := m_1^{s_1} m_2^{s_2}$ oraz $a_2 := g_1^{t_1} g_2^{t_2} h_2^{-d_2}$, $b_2 := m_1^{t_1} m_2^{t_2} h_2^{-d_2}$
3. P wysyła do V: (a_1, a_2, b_1, b_2)
4. V wysyła do P: $c \in_{\mathcal{R}} \mathbb{Z}_q^*$
5. P liczy $d_1 := c - d_2$, $(u_1, u_2) := (t_1, t_2)$ oraz $(r_1, r_2) := (s_1 + d_1 x_{11}, s_2 + d_1 x_{12})$
6. P wysyła do V: $(d_1, d_2, r_1, r_2, u_1, u_2)$

Protokół

Protokół

1. P: $s_1, s_2, t_1, t_2, d_2 \in_{\mathcal{R}} \mathbb{Z}_q^*$
2. P liczy $a_1 := g_1^{s_1} g_2^{s_2}$, $b_1 := m_1^{s_1} m_2^{s_2}$ oraz $a_2 := g_1^{t_1} g_2^{t_2} h_2^{-d_2}$, $b_2 := m_1^{t_1} m_2^{t_2} h_2^{-d_2}$
3. P wysyła do V: (a_1, a_2, b_1, b_2)
4. V wysyła do P: $c \in_{\mathcal{R}} \mathbb{Z}_q^*$
5. P liczy $d_1 := c - d_2$, $(u_1, u_2) := (t_1, t_2)$ oraz $(r_1, r_2) := (s_1 + d_1 x_{11}, s_2 + d_1 x_{12})$
6. P wysyła do V: $(d_1, d_2, r_1, r_2, u_1, u_2)$

Protokół

Protokół

1. P: $s_1, s_2, t_1, t_2, d_2 \in_{\mathcal{R}} \mathbb{Z}_q^*$
2. P liczy $a_1 := g_1^{s_1} g_2^{s_2}$, $b_1 := m_1^{s_1} m_2^{s_2}$ oraz $a_2 := g_1^{t_1} g_2^{t_2} h_2^{-d_2}$, $b_2 := m_1^{t_1} m_2^{t_2} h_2^{-d_2}$
3. P wysyła do V: (a_1, a_2, b_1, b_2)
4. V wysyła do P: $c \in_{\mathcal{R}} \mathbb{Z}_q^*$
5. P liczy $d_1 := c - d_2$, $(u_1, u_2) := (t_1, t_2)$ oraz $(r_1, r_2) := (s_1 + d_1 x_{11}, s_2 + d_1 x_{12})$
6. P wysyła do V: $(d_1, d_2, r_1, r_2, u_1, u_2)$

Protokół

Protokół

1. P: $s_1, s_2, t_1, t_2, d_2 \in_{\mathcal{R}} \mathbb{Z}_q^*$
2. P liczy $a_1 := g_1^{s_1} g_2^{s_2}$, $b_1 := m_1^{s_1} m_2^{s_2}$ oraz $a_2 := g_1^{t_1} g_2^{t_2} h_2^{-d_2}$, $b_2 := m_1^{t_1} m_2^{t_2} h_2^{-d_2}$
3. P wysyła do V: (a_1, a_2, b_1, b_2)
4. V wysyła do P: $c \in_{\mathcal{R}} \mathbb{Z}_q^*$
5. P liczy $d_1 := c - d_2$, $(u_1, u_2) := (t_1, t_2)$ oraz $(r_1, r_2) := (s_1 + d_1 x_{11}, s_2 + d_1 x_{12})$
6. P wysyła do V: $(d_1, d_2, r_1, r_2, u_1, u_2)$

Protokół

Protokół

1. P: $s_1, s_2, t_1, t_2, d_2 \in_{\mathcal{R}} \mathbb{Z}_q^*$
2. P *liczy* $a_1 := g_1^{s_1} g_2^{s_2}$, $b_1 := m_1^{s_1} m_2^{s_2}$ *oraz* $a_2 := g_1^{t_1} g_2^{t_2} h_2^{-d_2}$, $b_2 := m_1^{t_1} m_2^{t_2} h_2^{-d_2}$
3. P *wysyła do* V: (a_1, a_2, b_1, b_2)
4. V *wysyła do* P: $c \in_{\mathcal{R}} \mathbb{Z}_q^*$
5. P *liczy* $d_1 := c - d_2$, $(u_1, u_2) := (t_1, t_2)$ *oraz* $(r_1, r_2) := (s_1 + d_1 x_{11}, s_2 + d_1 x_{12})$
6. P *wysyła do* V: $(d_1, d_2, r_1, r_2, u_1, u_2)$

Protokół

Protokół

1. P: $s_1, s_2, t_1, t_2, d_2 \in_{\mathcal{R}} \mathbb{Z}_q^*$
2. P *liczy* $a_1 := g_1^{s_1} g_2^{s_2}$, $b_1 := m_1^{s_1} m_2^{s_2}$ *oraz* $a_2 := g_1^{t_1} g_2^{t_2} h_2^{-d_2}$, $b_2 := m_1^{t_1} m_2^{t_2} h_2^{-d_2}$
3. P *wysyła do* V: (a_1, a_2, b_1, b_2)
4. V *wysyła do* P: $c \in_{\mathcal{R}} \mathbb{Z}_q^*$
5. P *liczy* $d_1 := c - d_2$, $(u_1, u_2) := (t_1, t_2)$ *oraz* $(r_1, r_2) := (s_1 + d_1 x_{11}, s_2 + d_1 x_{12})$
6. P *wysyła do* V: $(d_1, d_2, r_1, r_2, u_1, u_2)$

Protokół, c.d.

Protokół (c.d. - weryfikacja)

V weryfikuje przesłane przez P wartości:

- $d_1 + d_2 \stackrel{?}{=} c$
- $g_1^{r_1} g_2^{r_2} \stackrel{?}{=} a_1 h_1^{d_1}$
- $m_1^{r_1} m_2^{r_2} \stackrel{?}{=} b_1 z^{d_1}$
- $g_1^{u_1} g_2^{u_2} \stackrel{?}{=} a_2 h_2^{d_2}$
- $m_1^{u_1} m_2^{u_2} \stackrel{?}{=} b_2 z^{d_2}$

Jak naprawdę tworzy się podpis

- $\mathcal{H}, \mathcal{H}_1, \mathcal{H}_2$ to funkcje haszujące, m – wiadomość
- $(m_1, m_2) := (\mathcal{H}_1(m), \mathcal{H}_2(m))$
- P oblicza z, $c := \mathcal{H}(a_1, b_1, a_2, b_2, m_1, m_2)$ i przeprowadza dowód.
- Podpisem pod m jest $(z, d_1, d_2, r_1, r_2, u_1, u_2)$.
- Weryfikacja polega na obliczeniu (a_1, b_1, a_2, b_2), sprawdzeniu że c ma poprawną wartość i że $c = d_1 + d_2$.

Jak naprawdę tworzy się podpis

- $\mathcal{H}, \mathcal{H}_1, \mathcal{H}_2$ to funkcje haszujące, m – wiadomość
- $(m_1, m_2) := (\mathcal{H}_1(m), \mathcal{H}_2(m))$
- P oblicza z, $c := \mathcal{H}(a_1, b_1, a_2, b_2, m_1, m_2)$ i przeprowadza dowód.
- Podpisem pod m jest $(z, d_1, d_2, r_1, r_2, u_1, u_2)$.
- Weryfikacja polega na obliczeniu (a_1, b_1, a_2, b_2), sprawdzeniu że c ma poprawną wartość i że $c = d_1 + d_2$.

Jak naprawdę tworzy się podpis

- $\mathcal{H}, \mathcal{H}_1, \mathcal{H}_2$ to funkcje haszujące, m – wiadomość
- $(m_1, m_2) := (\mathcal{H}_1(m), \mathcal{H}_2(m))$
- P oblicza z, $c := \mathcal{H}(a_1, b_1, a_2, b_2, m_1, m_2)$ i przeprowadza dowód.
- Podpisem pod m jest $(z, d_1, d_2, r_1, r_2, u_1, u_2)$.
- Weryfikacja polega na obliczeniu (a_1, b_1, a_2, b_2), sprawdzeniu że c ma poprawną wartość i że $c = d_1 + d_2$.

Jak naprawdę tworzy się podpis

- $\mathcal{H}, \mathcal{H}_1, \mathcal{H}_2$ to funkcje haszujące, m – wiadomość
- $(m_1, m_2) := (\mathcal{H}_1(m), \mathcal{H}_2(m))$
- P oblicza z, $c := \mathcal{H}(a_1, b_1, a_2, b_2, m_1, m_2)$ i przeprowadza dowód.
- Podpisem pod m jest $(z, d_1, d_2, r_1, r_2, u_1, u_2)$.
- Weryfikacja polega na obliczeniu (a_1, b_1, a_2, b_2), sprawdzeniu że c ma poprawną wartość i że $c = d_1 + d_2$.

Jak naprawdę tworzy się podpis

- $\mathcal{H}, \mathcal{H}_1, \mathcal{H}_2$ to funkcje haszujące, m – wiadomość
- $(m_1, m_2) := (\mathcal{H}_1(m), \mathcal{H}_2(m))$
- P oblicza z, $c := \mathcal{H}(a_1, b_1, a_2, b_2, m_1, m_2)$ i przeprowadza dowód.
- Podpisem pod m jest $(z, d_1, d_2, r_1, r_2, u_1, u_2)$.
- Weryfikacja polega na obliczeniu (a_1, b_1, a_2, b_2), sprawdzeniu że c ma poprawną wartość i że $c = d_1 + d_2$.

Własności

1. Nim P_i podpisze jakąkolwiek wiadomość, jest bezwarunkowo chroniony przed podrobieniem swojego podpisu. Po podpisaniu wiadomości reszta grupy może podrobić podpis i przy dostatecznie dużej mocy obliczeniowej.
2. Jeśli P_i podpisał 2 wiadomości, to przy odpowiednio dużej mocy obliczeniowej można stwierdzić, że obie wiadomości odpowiadają h_i (są podpisane przez P_i).
3. Jeśli P_1 i P_2 podpiszą po jednej wiadomości, to przy odpowiednio dużej mocy obliczeniowej można stwierdzić, że podpisy należą do różnych osób, ale nie da się stwierdzić których.

Własności

1. Nim P_i podpisze jakąkolwiek wiadomość, jest bezwarunkowo chroniony przed podrobieniem swojego podpisu. Po podpisaniu wiadomości reszta grupy może podrobić podpis *i* przy dostatecznie dużej mocy obliczeniowej.
2. Jeśli P_i podpisał 2 wiadomości, to przy odpowiednio dużej mocy obliczeniowej można stwierdzić, że obie wiadomości odpowiadają h_i (są podpisane przez P_i).
3. Jeśli P_1 i P_2 podpiszą po jednej wiadomości, to przy odpowiednio dużej mocy obliczeniowej można stwierdzić, że podpisy należą do różnych osób, ale nie da się stwierdzić których.

Własności

1. Nim P_i podpisze jakąkolwiek wiadomość, jest bezwarunkowo chroniony przed podrobieniem swojego podpisu. Po podpisaniu wiadomości reszta grupy może podrobić podpis i przy dostatecznie dużej mocy obliczeniowej.
2. Jeśli P_i podpisał 2 wiadomości, to przy odpowiednio dużej mocy obliczeniowej można stwierdzić, że obie wiadomości odpowiadają h_i (są podpisane przez P_i).
3. Jeśli P_1 i P_2 podpiszą po jednej wiadomości, to przy odpowiednio dużej mocy obliczeniowej można stwierdzić, że podpisy należą do różnych osób, ale nie da się stwierdzić których.

Własności

1. Nim P_i podpisze jakąkolwiek wiadomość, jest bezwarunkowo chroniony przed podrobieniem swojego podpisu. Po podpisaniu wiadomości reszta grupy może podrobić podpis i przy dostatecznie dużej mocy obliczeniowej.
2. Jeśli P_i podpisał 2 wiadomości, to przy odpowiednio dużej mocy obliczeniowej można stwierdzić, że obie wiadomości odpowiadają h_i (są podpisane przez P_i).
3. Jeśli P_1 i P_2 podpiszą po jednej wiadomości, to przy odpowiednio dużej mocy obliczeniowej można stwierdzić, że podpisy należą do różnych osób, ale nie da się stwierdzić których.

Podpisywanie wielu wiadomości

- Chcemy by każdy mógł podpisać l wiadomości.
- generatory G rzędu q $g_1, g_2, \ldots, g_{l+1}$
- klucz prywatny $P_i : (x_{i,1}, \ldots, x_{i,l+1}) \in \mathbb{Z}_q^{l+1}$
- klucz publiczny grupy: $(g_1, \ldots, g_{l+1}, h_1, h_2)$
  gdzie $h_i = g_1^{x_{i,1}} \cdots g_{l+1}^{x_{i,l+1}}$
- podpis P_i pod $m = (m_1, \ldots, m_{l+1}) : z = m_1^{x_{i,1}} \cdots m_{l+1}^{x_{i,l+1}}$
- Dla losowych M ppb, że macierz $t+1$ równań (t podpisów + klucz publiczny) ma rząd $t+1$ wynosi

$$\leq 1 - \sum_{j=1}^{t} \frac{q^j}{q^{l+1}} \approx 1 - \frac{q^t}{q^{l+1}}$$

Podpisywanie wielu wiadomości

- Chcemy by każdy mógł podpisać l wiadomości.
- generatory G rzędu q $g_1, g_2, \ldots, g_{l+1}$
- klucz prywatny $P_i : (x_{i,1}, \ldots, x_{i,l+1}) \in \mathbb{Z}_q^{l+1}$
- klucz publiczny grupy: $(g_1, \ldots, g_{l+1}, h_1, h_2)$
  gdzie $h_i = g_1^{x_{i,1}} \cdots g_{l+1}^{x_{i,l+1}}$
- podpis P_i pod $m = (m_1, \ldots, m_{l+1}) : z = m_1^{x_{i,1}} \cdots m_{l+1}^{x_{i,l+1}}$
- Dla losowych M ppb, że macierz $t+1$ równań (t podpisów + klucz publiczny) ma rząd $t+1$ wynosi

$$\leq 1 - \sum_{j=1}^{t} \frac{q^j}{q^{l+1}} \approx 1 - \frac{q^t}{q^{l+1}}$$

Podpisywanie wielu wiadomości

- Chcemy by każdy mógł podpisać l wiadomości.
- generatory G rzędu q $g_1, g_2, \ldots, g_{l+1}$
- klucz prywatny $P_i : (x_{i,1}, \ldots, x_{i,l+1}) \in \mathbb{Z}_q^{l+1}$
- klucz publiczny grupy: $(g_1, \ldots, g_{l+1}, h_1, h_2)$
  gdzie $h_i = g_1^{x_{i,1}} \cdots g_{l+1}^{x_{i,l+1}}$
- podpis P_i pod $m = (m_1, \ldots, m_{l+1}) : z = m_1^{x_{i,1}} \cdots m_{l+1}^{x_{i,l+1}}$
- Dla losowych M ppb, że macierz $t+1$ równań (t podpisów + klucz publiczny) ma rząd $t+1$ wynosi

$$\leq 1 - \sum_{j=1}^{t} \frac{q^j}{q^{l+1}} \approx 1 - \frac{q^t}{q^{l+1}}$$

Podpisywanie wielu wiadomości

- Chcemy by każdy mógł podpisać l wiadomości.
- generatory G rzędu q $g_1, g_2, \ldots, g_{l+1}$
- klucz prywatny $P_i : (x_{i,1}, \ldots, x_{i,l+1}) \in \mathbb{Z}_q^{l+1}$
- klucz publiczny grupy: $(g_1, \ldots, g_{l+1}, h_1, h_2)$
  gdzie $h_i = g_1^{x_{i,1}} \cdots g_{l+1}^{x_{i,l+1}}$
- podpis P_i pod $m = (m_1, \ldots, m_{l+1}) : z = m_1^{x_{i,1}} \cdots m_{l+1}^{x_{i,l+1}}$
- Dla losowych M ppb, że macierz $t+1$ równań (t podpisów + klucz publiczny) ma rząd $t+1$ wynosi

$$\leq 1 - \sum_{j=1}^{t} \frac{q^j}{q^{l+1}} \approx 1 - \frac{q^t}{q^{l+1}}$$

Podpisywanie wielu wiadomości

- Chcemy by każdy mógł podpisać l wiadomości.
- generatory G rzędu q $g_1, g_2, \ldots, g_{l+1}$
- klucz prywatny $P_i : (x_{i,1}, \ldots, x_{i,l+1}) \in \mathbb{Z}_q^{l+1}$
- klucz publiczny grupy: $(g_1, \ldots, g_{l+1}, h_1, h_2)$
  gdzie $h_i = g_1^{x_{i,1}} \cdots g_{l+1}^{x_{i,l+1}}$
- podpis P_i pod $m = (m_1, \ldots, m_{l+1}) : z = m_1^{x_{i,1}} \cdots m_{l+1}^{x_{i,l+1}}$
- Dla losowych M ppb, że macierz $t+1$ równań (t podpisów + klucz publiczny) ma rząd $t+1$ wynosi

$$\leq 1 - \sum_{j=1}^{t} \frac{q^j}{q^{l+1}} \approx 1 - \frac{q^t}{q^{l+1}}$$

Podpisywanie wielu wiadomości

- Chcemy by każdy mógł podpisać l wiadomości.
- generatory G rzędu q $g_1, g_2, \ldots, g_{l+1}$
- klucz prywatny $P_i : (x_{i,1}, \ldots, x_{i,l+1}) \in \mathbb{Z}_q^{l+1}$
- klucz publiczny grupy: $(g_1, \ldots, g_{l+1}, h_1, h_2)$
  gdzie $h_i = g_1^{x_{i,1}} \cdots g_{l+1}^{x_{i,l+1}}$
- podpis P_i pod $m = (m_1, \ldots, m_{l+1}) : z = m_1^{x_{i,1}} \cdots m_{l+1}^{x_{i,l+1}}$
- Dla losowych M ppb, że macierz $t+1$ równań (t podpisów + klucz publiczny) ma rząd $t+1$ wynosi

$$\leq 1 - \sum_{j=1}^{t} \frac{q^j}{q^{l+1}} \approx 1 - \frac{q^t}{q^{l+1}}$$

Schemat gwarantujący anonimowość obliczeniową

- Grupa znów składa się z 2 osób.
- klucz publiczny grupy: (g, h_1, h_2)
- klucz prywatny P_i: $x_i = log_g h_i$
- P_i podpisując m oblicza $z_i = m^{x_i}$ oraz losuje $z_{3-1} \in \mathbb{Z}_q$ i dowodzi, że zna w takie że:

$$(h_1 = g^w \wedge z_1 = m^w) \quad \vee \quad (h_2 = g^w \wedge z_2 = m^w)$$

Schemat gwarantujący anonimowość obliczeniową

- Grupa znów składa się z 2 osób.
- klucz publiczny grupy: (g, h_1, h_2)
- klucz prywatny P_i: $x_i = log_g h_i$
- P_i podpisując m oblicza $z_i = m^{x_i}$ oraz losuje $z_{3-1} \in \mathbb{Z}_q$ i dowodzi, że zna w takie że:

$$(h_1 = g^w \wedge z_1 = m^w) \quad \vee \quad (h_2 = g^w \wedge z_2 = m^w)$$

Schemat gwarantujący anonimowość obliczeniową

- Grupa znów składa się z 2 osób.
- klucz publiczny grupy: (g, h_1, h_2)
- klucz prywatny P_i: $x_i = log_g h_i$
- P_i podpisując m oblicza $z_i = m^{x_i}$ oraz losuje $z_{3-1} \in \mathbb{Z}_q$ i dowodzi, że zna w takie że:

$$(h_1 = g^w \wedge z_1 = m^w) \quad \vee \quad (h_2 = g^w \wedge z_2 = m^w)$$

Schemat gwarantujący anonimowość obliczeniową

- Grupa znów składa się z 2 osób.
- klucz publiczny grupy: (g, h_1, h_2)
- klucz prywatny P_i: $x_i = log_g h_i$
- P_i podpisując m oblicza $z_i = m^{x_i}$ oraz losuje $z_{3-1} \in \mathbb{Z}_q$ i dowodzi, że zna w takie że:

$$(h_1 = g^w \wedge z_1 = m^w) \quad \vee \quad (h_2 = g^w \wedge z_2 = m^w)$$

Protokół

Protokół

1. P: $s_1, s_2, d_2 \in_{\mathcal{R}} \mathbb{Z}_q^*$
2. P liczy $a_1 := g^{s_1}$, $a_2 := g^{s_2} h_2^{-d_2}$ oraz $b_1 := m^{s_1}$, $b_2 := m^{s_2} z_2^{-d_2}$
3. P wysyła do V: (a_1, a_2, b_1, b_2)
4. V wysyła do P: $c \in_{\mathcal{R}} \mathbb{Z}_q^*$
5. P liczy $d_1 := c - d_2$ oraz $(u_1, u_2) := (s_1 + d_1 x_1, s_2)$
6. P wysyła do V: (d_1, d_2, u_1, u_2)

Protokół

Protokół

1. P: $s_1, s_2, d_2 \in_{\mathcal{R}} \mathbb{Z}_q^*$
2. P liczy $a_1 := g^{s_1}$, $a_2 := g^{s_2} h_2^{-d_2}$ oraz $b_1 := m^{s_1}$, $b_2 := m^{s_2} z_2^{-d_2}$
3. P wysyła do V: (a_1, a_2, b_1, b_2)
4. V wysyła do P: $c \in_{\mathcal{R}} \mathbb{Z}_q^*$
5. P liczy $d_1 := c - d_2$ oraz $(u_1, u_2) := (s_1 + d_1 x_1, s_2)$
6. P wysyła do V: (d_1, d_2, u_1, u_2)

Protokół

Protokół

1. P: $s_1, s_2, d_2 \in_{\mathcal{R}} \mathbb{Z}_q^*$
2. P *liczy* $a_1 := g^{s_1}$, $a_2 := g^{s_2} h_2^{-d_2}$ *oraz* $b_1 := m^{s_1}$, $b_2 := m^{s_2} z_2^{-d_2}$
3. P *wysyła do* V: (a_1, a_2, b_1, b_2)
4. V *wysyła do* P: $c \in_{\mathcal{R}} \mathbb{Z}_q^*$
5. P *liczy* $d_1 := c - d_2$ *oraz* $(u_1, u_2) := (s_1 + d_1 x_1, s_2)$
6. P *wysyła do* V: (d_1, d_2, u_1, u_2)

Protokół

Protokół

1. P: $s_1, s_2, d_2 \in_{\mathcal{R}} \mathbb{Z}_q^*$
2. P *liczy* $a_1 := g^{s_1}$, $a_2 := g^{s_2} h_2^{-d_2}$ *oraz* $b_1 := m^{s_1}$, $b_2 := m^{s_2} z_2^{-d_2}$
3. P *wysyła do* V: (a_1, a_2, b_1, b_2)
4. V *wysyła do* P: $c \in_{\mathcal{R}} \mathbb{Z}_q^*$
5. P *liczy* $d_1 := c - d_2$ *oraz* $(u_1, u_2) := (s_1 + d_1 x_1, s_2)$
6. P *wysyła do* V: (d_1, d_2, u_1, u_2)

Protokół

Protokół

1. P: $s_1, s_2, d_2 \in_{\mathcal{R}} \mathbb{Z}_q^*$
2. P *liczy* $a_1 := g^{s_1}$, $a_2 := g^{s_2} h_2^{-d_2}$ *oraz* $b_1 := m^{s_1}$, $b_2 := m^{s_2} z_2^{-d_2}$
3. P *wysyła do* V: (a_1, a_2, b_1, b_2)
4. V *wysyła do* P: $c \in_{\mathcal{R}} \mathbb{Z}_q^*$
5. P *liczy* $d_1 := c - d_2$ *oraz* $(u_1, u_2) := (s_1 + d_1 x_1, s_2)$
6. P *wysyła do* V: (d_1, d_2, u_1, u_2)

Protokół

Protokół

1. P: $s_1, s_2, d_2 \in_{\mathcal{R}} \mathbb{Z}_q^*$
2. P *liczy* $a_1 := g^{s_1}$, $a_2 := g^{s_2} h_2^{-d_2}$
   oraz $b_1 := m^{s_1}$, $b_2 := m^{s_2} z_2^{-d_2}$
3. P *wysyła do* V: (a_1, a_2, b_1, b_2)
4. V *wysyła do* P: $c \in_{\mathcal{R}} \mathbb{Z}_q^*$
5. P *liczy* $d_1 := c - d_2$ *oraz* $(u_1, u_2) := (s_1 + d_1 x_1, s_2)$
6. P *wysyła do* V: (d_1, d_2, u_1, u_2)

Protokół

Protokół

1. P: $s_1, s_2, d_2 \in_{\mathcal{R}} \mathbb{Z}_q^*$
2. P *liczy* $a_1 := g^{s_1}$, $a_2 := g^{s_2} h_2^{-d_2}$
   oraz $b_1 := m^{s_1}$, $b_2 := m^{s_2} z_2^{-d_2}$
3. P *wysyła do* V: (a_1, a_2, b_1, b_2)
4. V *wysyła do* P: $c \in_{\mathcal{R}} \mathbb{Z}_q^*$
5. P *liczy* $d_1 := c - d_2$ *oraz* $(u_1, u_2) := (s_1 + d_1 x_1, s_2)$
6. P *wysyła do* V: (d_1, d_2, u_1, u_2)

Protokół, c.d.

Protokół (c.d. - weryfikacja)

V weryfikuje przesłane przez P wartości:

- $d_1 + d_2 \stackrel{?}{=} c$
- $g^{u_1} \stackrel{?}{=} a_1 h_1^{d_1}$
- $m^{u_1} \stackrel{?}{=} b_1 z_1^{d_1}$
- $g^{u_2} \stackrel{?}{=} a_2 h_2^{d_2}$
- $m^{u_2} \stackrel{?}{=} b_2 z_2^{d_2}$

Własności

Schemat posiada poniższe własności przy pewnych rozsądnych założeniach dot. mocy obliczeniowej.

- Mając daną wiadomość m podpisaną przez P_1 lub P_2 „nie da się” stwierdzić, kto dokładnie ją podpisał.
- Co jeśli dysponujemy pewną liczbą podpisanych wiadomości wraz z informacją, kto złożył podpis?

Własności

Schemat posiada poniższe własności przy pewnych rozsądnych założeniach dot. mocy obliczeniowej.

- Mając daną wiadomość m podpisaną przez P_1 lub P_2 „nie da się” stwierdzić, kto dokładnie ją podpisał.
- Co jeśli dysponujemy pewną liczbą podpisanych wiadomości wraz z informacją, kto złożył podpis?

Własności

Schemat posiada poniższe własności przy pewnych rozsądnych założeniach dot. mocy obliczeniowej.

- Mając daną wiadomość m podpisaną przez P_1 lub P_2 „nie da się” stwierdzić, kto dokładnie ją podpisał.
- Co jeśli dysponujemy pewną liczbą podpisanych wiadomości wraz z informacją, kto złożył podpis?

Własności, cd.

- Załóżmy, że złożenie podpisu jest równoważne przeprowadzeniu protokołu (tj. że wynik $\mathcal{H}$ odpowiada wylosowaniu pewnej liczby).
- Znając tożsamość dowodzącego, sami potrafimy generować zapisy tego protokołu, z tym samym ppb.!
- $\approx$ Potrafimy generować wiadomości podpisane przez dowolnego członka grupy – czyli ich znajomość nic nam nie daje.
- Jedyne, co uzyskujemy z poprzednich podpisów to wartości $m_i^{x_j}$ dla $i = 1, 2, \ldots$ oraz $j = 1$ lub $j = 2$
- Znów z „rozsądnego założenia” – te informacje nic nie dają!

Własności, cd.

- Załóżmy, że złożenie podpisu jest równoważne przeprowadzeniu protokołu (tj. że wynik $\mathcal{H}$ odpowiada wylosowaniu pewnej liczby).
- Znając tożsamość dowodzącego, sami potrafimy generować zapisy tego protokołu, z tym samym ppb.!
- $\approx$ Potrafimy generować wiadomości podpisane przez dowolnego członka grupy – czyli ich znajomość nic nam nie daje.
- Jedyne, co uzyskujemy z poprzednich podpisów to wartości $m_i^{x_j}$ dla $i = 1, 2, \ldots$ oraz $j = 1$ lub $j = 2$
- Znów z „rozsądnego założenia” – te informacje nic nie dają!

Własności, cd.

- Załóżmy, że złożenie podpisu jest równoważne przeprowadzeniu protokołu (tj. że wynik $\mathcal{H}$ odpowiada wylosowaniu pewnej liczby).
- Znając tożsamość dowodzącego, sami potrafimy generować zapisy tego protokołu, z tym samym ppb.!
- $\approx$ Potrafimy generować wiadomości podpisane przez dowolnego członka grupy – czyli ich znajomość nic nam nie daje.
- Jedyne, co uzyskujemy z poprzednich podpisów to wartości $m_i^{x_j}$ dla $i = 1, 2, \ldots$ oraz $j = 1$ lub $j = 2$
- Znów z „rozsądnego założenia” – te informacje nic nie dają!

Własności, cd.

- Załóżmy, że złożenie podpisu jest równoważne przeprowadzeniu protokołu (tj. że wynik $\mathcal{H}$ odpowiada wylosowaniu pewnej liczby).
- Znając tożsamość dowodzącego, sami potrafimy generować zapisy tego protokołu, z tym samym ppb.!
- $\approx$ Potrafimy generować wiadomości podpisane przez dowolnego członka grupy – czyli ich znajomość nic nam nie daje.
- Jedyne, co uzyskujemy z poprzednich podpisów to wartości $m_i^{x_j}$ dla $i = 1, 2, \ldots$ oraz $j = 1$ lub $j = 2$
- Znów z „rozsądnego założenia” – te informacje nic nie dają!

Własności, cd.

- Załóżmy, że złożenie podpisu jest równoważne przeprowadzeniu protokołu (tj. że wynik $\mathcal{H}$ odpowiada wylosowaniu pewnej liczby).
- Znając tożsamość dowodzącego, sami potrafimy generować zapisy tego protokołu, z tym samym ppb.!
- $\approx$ Potrafimy generować wiadomości podpisane przez dowolnego członka grupy – czyli ich znajomość nic nam nie daje.
- Jedyne, co uzyskujemy z poprzednich podpisów to wartości $m_i^{x_j}$ dla $i = 1, 2, \ldots$ oraz $j = 1$ lub $j = 2$
- Znów z „rozsądnego założenia” – te informacje nic nie dają!

Własności, cd.

- Załóżmy, że złożenie podpisu jest równoważne przeprowadzeniu protokołu (tj. że wynik $\mathcal{H}$ odpowiada wylosowaniu pewnej liczby).
- Znając tożsamość dowodzącego, sami potrafimy generować zapisy tego protokołu, z tym samym ppb.!
- $\approx$ Potrafimy generować wiadomości podpisane przez dowolnego członka grupy – czyli ich znajomość nic nam nie daje.
- Jedyne, co uzyskujemy z poprzednich podpisów to wartości $m_i^{x_j}$ dla $i = 1, 2, \ldots$ oraz $j = 1$ lub $j = 2$
- Znów z „rozsądnego założenia” – te informacje nic nie dają!

Założenia dotyczące bezpieczeństwa

Przypuszczenie

1. *Strong RSA Assumption*
2. *Modified Strong RSA Assumption*
3. *Diffie-Hellman Decision Assumption*

Następne schematy podpisów mają kilka cech wspólnych:

- są adaptacjami protokołów dowodu wiedzy
- używają funkcji haszujących ($\mathcal{H}\colon \{0,1\}^* \to \{0,1\}^k$)
- podpisem jest (m.in.) c takie, że $c = \mathcal{H}(f(c))$
- przy czym f jest funkcją wielu zmiennych – także wiadomości m oraz pozostałych elementów podpisu
- P nie zna rzędu grupy!
- stosujemy notację, której przykładem jest $SPK\{(\alpha,\beta) : \quad y = g^\alpha \wedge z = g^\beta h^\alpha\}(m)$

Następne schematy podpisów mają kilka cech wspólnych:

- są adaptacjami protokołów dowodu wiedzy
- używają funkcji haszujących ($\mathcal{H}\colon \{0,1\}^* \to \{0,1\}^k$)
- podpisem jest (m.in.) c takie, że $c = \mathcal{H}(f(c))$
- przy czym f jest funkcją wielu zmiennych – także wiadomości m oraz pozostałych elementów podpisu
- P nie zna rzędu grupy!
- stosujemy notację, której przykładem jest $SPK\{(\alpha,\beta) : \quad y = g^\alpha \wedge z = g^\beta h^\alpha\}(m)$

Następne schematy podpisów mają kilka cech wspólnych:

- są adaptacjami protokołów dowodu wiedzy
- używają funkcji haszujących ($\mathcal{H}\colon \{0,1\}^* \to \{0,1\}^k$)
- podpisem jest (m.in.) c takie, że $c = \mathcal{H}(f(c))$
- przy czym f jest funkcją wielu zmiennych – także wiadomości m oraz pozostałych elementów podpisu
- P nie zna rzędu grupy!
- stosujemy notację, której przykładem jest $SPK\{(\alpha,\beta) : \quad y = g^\alpha \wedge z = g^\beta h^\alpha\}(m)$

Następne schematy podpisów mają kilka cech wspólnych:

- są adaptacjami protokołów dowodu wiedzy
- używają funkcji haszujących ($\mathcal{H}\colon \{0,1\}^* \to \{0,1\}^k$)
- podpisem jest (m.in.) c takie, że $c = \mathcal{H}(f(c))$
- przy czym f jest funkcją wielu zmiennych – także wiadomości m oraz pozostałych elementów podpisu
- P nie zna rzędu grupy!
- stosujemy notację, której przykładem jest $SPK\{(\alpha,\beta) : \quad y = g^\alpha \wedge z = g^\beta h^\alpha\}(m)$

Następne schematy podpisów mają kilka cech wspólnych:

- są adaptacjami protokołów dowodu wiedzy
- używają funkcji haszujących ($\mathcal{H}\colon \{0,1\}^* \to \{0,1\}^k$)
- podpisem jest (m.in.) c takie, że $c = \mathcal{H}(f(c))$
- przy czym f jest funkcją wielu zmiennych – także wiadomości m oraz pozostałych elementów podpisu
- P nie zna rzędu grupy!
- stosujemy notację, której przykładem jest $SPK\{(\alpha,\beta) : \quad y = g^\alpha \wedge z = g^\beta h^\alpha\}(m)$

Następne schematy podpisów mają kilka cech wspólnych:

- są adaptacjami protokołów dowodu wiedzy
- używają funkcji haszujących ($\mathcal{H}\colon \{0,1\}^* \to \{0,1\}^k$)
- podpisem jest (m.in.) c takie, że $c = \mathcal{H}(f(c))$
- przy czym f jest funkcją wielu zmiennych – także wiadomości m oraz pozostałych elementów podpisu
- P nie zna rzędu grupy!
- stosujemy notację, której przykładem jest $SPK\{(\alpha,\beta) : \quad y = g^\alpha \wedge z = g^\beta h^\alpha\}(m)$

Następne schematy podpisów mają kilka cech wspólnych:

- są adaptacjami protokołów dowodu wiedzy
- używają funkcji haszujących ($\mathcal{H}\colon \{0,1\}^* \to \{0,1\}^k$)
- podpisem jest (m.in.) c takie, że $c = \mathcal{H}(f(c))$
- przy czym f jest funkcją wielu zmiennych – także wiadomości m oraz pozostałych elementów podpisu
- P nie zna rzędu grupy!
- stosujemy notację, której przykładem jest $SPK\{(\alpha,\beta) : \quad y = g^\alpha \wedge z = g^\beta h^\alpha\}(m)$

Logarytm dyskretny

Definicja

- $\epsilon > 1$ – *parametr bezpieczeństwa*
- *podpis pod m względem y:*
  $(c, s) \in \{0,1\}^k \times \{-2^{l_g+k}, \ldots, 2^{\epsilon(l_g+k)}\}$ *takie, że*
  $c = \mathcal{H}(g||y||g^s y^c||m)$
- *podpis ten oznaczamy* $SPK\{(\alpha) : \quad y = g^\alpha\}(m)$

Znając $x \in \{0,1\}^{l_g}$ takie, że $x = \log_g y$ można podpisać tak:

- wylosować $r \in_{\mathcal{R}} \{0,1\}^{\epsilon(l_g+k)}$ i policzyć $t := g^r$
- $c := \mathcal{H}(g||y||t||m)$
- $s := r - cx$ (w $\mathbb{Z}$)

Logarytm dyskretny

Definicja

- $\epsilon > 1$ – *parametr bezpieczeństwa*
- *podpis pod m względem y:*
  $(c, s) \in \{0,1\}^k \times \{-2^{l_g+k}, \ldots, 2^{\epsilon(l_g+k)}\}$ *takie, że*
  $c = \mathcal{H}(g||y||g^s y^c||m)$
- *podpis ten oznaczamy* $SPK\{(\alpha) : \quad y = g^\alpha\}(m)$

Znając $x \in \{0,1\}^{l_g}$ takie, że $x = \log_g y$ można podpisać tak:

- wylosować $r \in_{\mathcal{R}} \{0,1\}^{\epsilon(l_g+k)}$ i policzyć $t := g^r$
- $c := \mathcal{H}(g||y||t||m)$
- $s := r - cx$ (w $\mathbb{Z}$)

Logarytm dyskretny, cd.

Lemat

Pod warunkiem (1), interaktywny protokół odpowiadający $SPK\{(\alpha) : \quad y = g^{\alpha}\}(m)$ *jest dowodem ze statystyczną wiedzą zerową przy uczciwym weryfikatorze znajomości logarytmu dyskretnego z y.*

Równość dwóch logarytmów dyskretnych

Definicja

- $\epsilon > 1$ – *parametr bezpieczeństwa*
- *podpis pod m względem* y_1, y_2*:*
  $(c, s) \in \{0,1\}^k \times \{-2^{l_g+k}, \ldots, 2^{\epsilon(l_g+k)}\}$ *takie, że*
  $c = \mathcal{H}(g||h||y_1||y_2||y_1^c g^s||y_2^c h^s||m)$
- *podpis ten oznaczamy*
  $SPK\{(\alpha) : \quad y_1 = g^\alpha \wedge \quad y_2 = h^\alpha\}(m)$

Znając $x \in \{0,1\}^{l_g}$: $y_1 = g^x$ i $y_2 = h^x$ można podpisać tak:

- wylosować $r \in_{\mathcal{R}} \{0,1\}^{\epsilon(l_g+k)}$ i policzyć $t_1 := g^r$, $t_2 := h^r$
- $c := \mathcal{H}(g||h||y_1||y_2||t_1||t_2||m)$
- $s := r - cx$ (w $\mathbb{Z}$)

Równość dwóch logarytmów dyskretnych

Definicja

- $\epsilon > 1$ – *parametr bezpieczeństwa*
- *podpis pod m względem* y_1, y_2*:*
  $(c, s) \in \{0,1\}^k \times \{-2^{l_g+k}, \ldots, 2^{\epsilon(l_g+k)}\}$ *takie, że*
  $c = \mathcal{H}(g||h||y_1||y_2||y_1^c g^s||y_2^c h^s||m)$
- *podpis ten oznaczamy*
  $SPK\{(\alpha) : \quad y_1 = g^\alpha \wedge \quad y_2 = h^\alpha\}(m)$

Znając $x \in \{0,1\}^{l_g}$: $y_1 = g^x$ i $y_2 = h^x$ można podpisać tak:

- wylosować $r \in_{\mathcal{R}} \{0,1\}^{\epsilon(l_g+k)}$ i policzyć $t_1 := g^r$, $t_2 := h^r$
- $c := \mathcal{H}(g||h||y_1||y_2||t_1||t_2||m)$
- $s := r - cx$ (w $\mathbb{Z}$)

Jeden z dwóch logarytmów dyskretnych

Definicja

- $\epsilon > 1$ – *parametr bezpieczeństwa*
- *podpis pod m względem* y_1, y_2*:*
  $(c_1, c_2, s_1, s_2) \in (\{0,1\}^k)^2 \times (\{-2^{l_g+k}, \ldots, 2^{\epsilon(l_g+k)}\})^2$ *takie, że* $c_1 \oplus c_2 = \mathcal{H}(g||h||y_1||y_2||y_1^{c_1} g^{s_1}||y_2^{c_2} h^{s_2}||m)$
- *podpis ten oznaczamy*
  $SPK\{(\alpha, \beta) : \quad y_1 = g^\alpha \vee y_2 = h^\beta\}(m)$

Znając $x \in \{0,1\}^{l_g}$ takie, że $y_1 = g^x$ można podpisać tak:

- wylosować $r_1, r_2 \in_{\mathcal{R}} \{0,1\}^{\epsilon(l_g+k)}$, $c_2 \in \{0,1\}^k$ i policzyć $t_1 := g^{r_1}$, $t_2 := h^{r_2} y_2^{c_2}$
- $c_1 := c_2 \oplus \mathcal{H}(g||h||y_1||y_2||t_1||t_2||m)$
- $s_1 := r_1 - cx$ (w $\mathbb{Z}$), $s_2 := r_2$

Jeden z dwóch logarytmów dyskretnych

Definicja

- $\epsilon > 1$ – *parametr bezpieczeństwa*
- *podpis pod m względem* y_1, y_2*:*
  $(c_1, c_2, s_1, s_2) \in (\{0,1\}^k)^2 \times (\{-2^{l_g+k}, \ldots, 2^{\epsilon(l_g+k)}\})^2$ *takie, że* $c_1 \oplus c_2 = \mathcal{H}(g||h||y_1||y_2||y_1^{c_1} g^{s_1}||y_2^{c_2} h^{s_2}||m)$
- *podpis ten oznaczamy*
  $SPK\{(\alpha, \beta) : \quad y_1 = g^\alpha \vee y_2 = h^\beta\}(m)$

Znając $x \in \{0,1\}^{l_g}$ takie, że $y_1 = g^x$ można podpisać tak:

- wylosować $r_1, r_2 \in_{\mathcal{R}} \{0,1\}^{\epsilon(l_g+k)}$, $c_2 \in \{0,1\}^k$ i policzyć $t_1 := g^{r_1}$, $t_2 := h^{r_2} y_2^{c_2}$
- $c_1 := c_2 \oplus \mathcal{H}(g||h||y_1||y_2||t_1||t_2||m)$
- $s_1 := r_1 - cx$ (w $\mathbb{Z}$), $s_2 := r_2$

Logarytm dyskretny w przedziale

Definicja

- $\epsilon > 1$ – *parametr bezpieczeństwa,* $l_1 < l_g$, l_2 – *długości*
- *podpis pod m względem y:*
  $(c, s) \in \{0,1\}^k \times \{-2^{l_g+k}, \ldots, 2^{\epsilon(l_g+k)}\}$ *takie, że*
  $c = \mathcal{H}(g||y||g^{s-c2^{l_2+k}} y^c||m)$
- *podpis ten oznaczamy* $SPK\{(\alpha) : \quad y = g^\alpha \wedge 2^{l_1} - 2^{\epsilon(l_g+k)+1} < \alpha < 2^{l_1} + 2^{\epsilon(l_g+k)+1}\}(m)$

Znając $x \in \{2^{l_1}, \ldots, 2^{l_1} + 2^{l_2}\}$: $y = g^x$ można podpisać tak:

- wylosować $r \in_{\mathcal{R}} \{0,1\}^{\epsilon(l_2+k)}$ i policzyć $t := g^r$
- $c := \mathcal{H}(g||y||t||m)$
- $s := r - c(x - 2^{l_1})$ (w $\mathbb{Z}$)

Logarytm dyskretny w przedziale

Definicja

- $\epsilon > 1$ – *parametr bezpieczeństwa,* $l_1 < l_g$, l_2 – *długości*
- *podpis pod m względem y:*
  $(c, s) \in \{0,1\}^k \times \{-2^{l_g+k}, \ldots, 2^{\epsilon(l_g+k)}\}$ *takie, że*
  $c = \mathcal{H}(g||y||g^{s-c2^{l_2+k}} y^c||m)$
- *podpis ten oznaczamy* $SPK\{(\alpha) : \quad y = g^\alpha \wedge 2^{l_1} - 2^{\epsilon(l_g+k)+1} < \alpha < 2^{l_1} + 2^{\epsilon(l_g+k)+1}\}(m)$

Znając $x \in \{2^{l_1}, \ldots, 2^{l_1} + 2^{l_2}\}$: $y = g^x$ można podpisać tak:

- wylosować $r \in_{\mathcal{R}} \{0,1\}^{\epsilon(l_2+k)}$ i policzyć $t := g^r$
- $c := \mathcal{H}(g||y||t||m)$
- $s := r - c(x - 2^{l_1})$ (w $\mathbb{Z}$)

Logarytm dyskretny w przedziale, cd.

Lemat

Pod warunkiem (1), interaktywny protokół odpowiadający $SPK\{(\alpha) : \quad y = g^{\alpha} \wedge 2^{l_1} - 2^{\epsilon(l_g+k)+1} < \alpha < 2^{l_1} + 2^{\epsilon(l_g+k)+1}\}(m)$ *jest dowodem ze statystyczną wiedzą zerową przy uczciwym weryfikatorze znajomości liczby całkowitej* x *takiej, że* $x \in \{2^{l_1}, \ldots, 2^{l_1} + 2^{l_2}\}$ *oraz* $y = g^x$.

Krótko o schemacie

- Moc TA jest rozdzielona pomiędzy 2 osoby:
  - MM – odpowiada za utworzenie grupy i dodawanie do niej kolejnych członków
  - RM – odpowiada za otwieranie podpisów
- bezpieczeństwo oparte o założenia (2) i (3)
- MM wybiera grupę $G = \langle g \rangle$ oraz kolejne generatory: z i h
- tylko MM zna rząd grupy G, ale wszyscy znają rząd wielkości: 2^{l_g}
- RM wybiera klucz prywatny x i publikuje publiczny $y = g^x$
- P_i (łącznie z MM) losuje klucz prywatny – l. pierwszą e z określonego przedziału
- P_i dostaje od MM certyfika przynależności do grupy: $u \in G$ takie, że $u^e = z$. MM nie może poznać e!

Krótko o schemacie

- Moc TA jest rozdzielona pomiędzy 2 osoby:
  - MM – odpowiada za utworzenie grupy i dodawanie do niej kolejnych członków
  - RM – odpowiada za otwieranie podpisów
- bezpieczeństwo oparte o założenia (2) i (3)
- MM wybiera grupę $G = \langle g \rangle$ oraz kolejne generatory: z i h
- tylko MM zna rząd grupy G, ale wszyscy znają rząd wielkości: 2^{l_g}
- RM wybiera klucz prywatny x i publikuje publiczny $y = g^x$
- P_i (łącznie z MM) losuje klucz prywatny – l. pierwszą e z określonego przedziału
- P_i dostaje od MM certyfika przynależności do grupy: $u \in G$ takie, że $u^e = z$. MM nie może poznać e!

Krótko o schemacie

- Moc TA jest rozdzielona pomiędzy 2 osoby:
  - MM – odpowiada za utworzenie grupy i dodawanie do niej kolejnych członków
  - RM – odpowiada za otwieranie podpisów
- bezpieczeństwo oparte o założenia (2) i (3)
- MM wybiera grupę $G = \langle g \rangle$ oraz kolejne generatory: z i h
- tylko MM zna rząd grupy G, ale wszyscy znają rząd wielkości: 2^{l_g}
- RM wybiera klucz prywatny x i publikuje publiczny $y = g^x$
- P_i (łącznie z MM) losuje klucz prywatny – l. pierwszą e z określonego przedziału
- P_i dostaje od MM certyfika przynależności do grupy: $u \in G$ takie, że $u^e = z$. MM nie może poznać e!

Krótko o schemacie

- Moc TA jest rozdzielona pomiędzy 2 osoby:
  - MM – odpowiada za utworzenie grupy i dodawanie do niej kolejnych członków
  - RM – odpowiada za otwieranie podpisów
- bezpieczeństwo oparte o założenia (2) i (3)
- MM wybiera grupę $G = \langle g \rangle$ oraz kolejne generatory: z i h
- tylko MM zna rząd grupy G, ale wszyscy znają rząd wielkości: 2^{l_g}
- RM wybiera klucz prywatny x i publikuje publiczny $y = g^x$
- P_i (łącznie z MM) losuje klucz prywatny – l. pierwszą e z określonego przedziału
- P_i dostaje od MM certyfika przynależności do grupy: $u \in G$ takie, że $u^e = z$. MM nie może poznać e!

Krótko o schemacie

- Moc TA jest rozdzielona pomiędzy 2 osoby:
  - MM – odpowiada za utworzenie grupy i dodawanie do niej kolejnych członków
  - RM – odpowiada za otwieranie podpisów
- bezpieczeństwo oparte o założenia (2) i (3)
- MM wybiera grupę $G = \langle g \rangle$ oraz kolejne generatory: z i h
- tylko MM zna rząd grupy G, ale wszyscy znają rząd wielkości: 2^{l_g}
- RM wybiera klucz prywatny x i publikuje publiczny $y = g^x$
- P_i (łącznie z MM) losuje klucz prywatny – l. pierwszą e z określonego przedziału
- P_i dostaje od MM certyfika przynależności do grupy: $u \in G$ takie, że $u^e = z$. MM nie może poznać e!

Krótko o schemacie

- Moc TA jest rozdzielona pomiędzy 2 osoby:
  - MM – odpowiada za utworzenie grupy i dodawanie do niej kolejnych członków
  - RM – odpowiada za otwieranie podpisów
- bezpieczeństwo oparte o założenia (2) i (3)
- MM wybiera grupę $G = \langle g \rangle$ oraz kolejne generatory: z i h
- tylko MM zna rząd grupy G, ale wszyscy znają rząd wielkości: 2^{l_g}
- RM wybiera klucz prywatny x i publikuje publiczny $y = g^x$
- P_i (łącznie z MM) losuje klucz prywatny – l. pierwszą e z określonego przedziału
- P_i dostaje od MM certyfika przynależności do grupy: $u \in G$ takie, że $u^e = z$. MM nie może poznać e!

Krótko o schemacie

- Moc TA jest rozdzielona pomiędzy 2 osoby:
  - MM – odpowiada za utworzenie grupy i dodawanie do niej kolejnych członków
  - RM – odpowiada za otwieranie podpisów
- bezpieczeństwo oparte o założenia (2) i (3)
- MM wybiera grupę $G = \langle g \rangle$ oraz kolejne generatory: z i h
- tylko MM zna rząd grupy G, ale wszyscy znają rząd wielkości: 2^{l_g}
- RM wybiera klucz prywatny x i publikuje publiczny $y = g^x$
- P_i (łącznie z MM) losuje klucz prywatny – l. pierwszą e z określonego przedziału
- P_i dostaje od MM certyfika przynależności do grupy: $u \in G$ takie, że $u^e = z$. MM nie może poznać e!

Podpis

Podpisem pod wiadomością m jest $(a, b, d) \in G^3$ wraz z SPK liczb u oraz e takie, że

- (a,b) jest zaszyfrowaną przy użyciu klucza publicznego RM postacią u
- d jest zobowiązaniem do e
- e leży w określonym przedziale
- $u^e = z$
- tożsamość osoby, która złożyła podpis może wyjawić RM, odszyfrowując (a, b)

Podpis

Podpisem pod wiadomością m jest $(a, b, d) \in G^3$ wraz z SPK liczb u oraz e takie, że

- (a,b) jest zaszyfrowaną przy użyciu klucza publicznego RM postacią u
- d jest zobowiązaniem do e
- e leży w określonym przedziale
- $u^e = z$
- tożsamość osoby, która złożyła podpis może wyjawić RM, odszyfrowując (a, b)

Podpis

Podpisem pod wiadomością m jest $(a, b, d) \in G^3$ wraz z SPK liczb u oraz e takie, że

- (a,b) jest zaszyfrowaną przy użyciu klucza publicznego RM postacią u
- d jest zobowiązaniem do e
- e leży w określonym przedziale
- $u^e = z$
- tożsamość osoby, która złożyła podpis może wyjawić RM, odszyfrowując (a, b)

Podpis

Podpisem pod wiadomością m jest $(a, b, d) \in G^3$ wraz z SPK liczb u oraz e takie, że

- (a,b) jest zaszyfrowaną przy użyciu klucza publicznego RM postacią u
- d jest zobowiązaniem do e
- e leży w określonym przedziale
- $u^e = z$
- tożsamość osoby, która złożyła podpis może wyjawić RM, odszyfrowując (a, b)

Podpis

Podpisem pod wiadomością m jest $(a, b, d) \in G^3$ wraz z SPK liczb u oraz e takie, że

- (a,b) jest zaszyfrowaną przy użyciu klucza publicznego RM postacią u
- d jest zobowiązaniem do e
- e leży w określonym przedziale
- $u^e = z$
- tożsamość osoby, która złożyła podpis może wyjawić RM, odszyfrowując (a, b)

Podpis

Podpisem pod wiadomością m jest $(a, b, d) \in G^3$ wraz z SPK liczb u oraz e takie, że

- (a,b) jest zaszyfrowaną przy użyciu klucza publicznego RM postacią u
- d jest zobowiązaniem do e
- e leży w określonym przedziale
- $u^e = z$
- tożsamość osoby, która złożyła podpis może wyjawić RM, odszyfrowując (a, b)

Zalety schematu

Długość klucza publicznego grupy oraz długość podpisu pod wiadomością nie zależą od rozmiaru grupy!