# Autoreferat[1]

## 1 Imię i Nazwisko

Łukasz Krzywiecki

## 2 Posiadane dyplomy i stopnie naukowe

**Stopień naukowy doktora:**

- Doktor nauk technicznych w zakresie informatyki (dyscyplina **informatyka**); Wydział Informatyki i Zarządzania, Politechnika Wrocławska; tytuł rozprawy: *Teoriogrowe algorytmy wielokryterialnej optymalizacji wybranych parametrów sieci komputerowej*.

Ponadto uzyskałem **tytuł zawodowy magistra inżyniera:**

- Magister inżynier (**informatyka**, specjalność: systemy informacji naukowo-technicznej); Wydział Informatyki i Zarządzania, Politechnika Wrocławska; obrona 25 marca 1997r.

## 3 Dotychczasowe zatrudnieniu w jednostkach naukowych

- Politechnika Wrocławska, Instytut Matematyki i Informatyki. Od 01.03.2003 jako asystent. Od 01.10.2004 jako adiunkt.
- Politechnika Wrocławska, Katedry Informatyki Wydziału Podstawowych Problemów Techniki. Od 01.10.2015 jako adiunkt.

## 4 Osiągnięcie naukowe

### 4.1 Tytuł osiągnięcia

Osiągnięciem jest jednotematyczny cykl publikacji zatytułowany:

*Protokoły ustalania kryptograficznych kluczy sesyjnych wspierające ochronę prywatności.*

---

[1] Format dokumentu na podstawie wzoru umieszczonego na stronach CK oraz § 4 i 5 Rozporządzenia Ministra Nauki i Szkolnictwa Wyższego z dnia 1 września 2011 (Dz. Ustaw 196 poz. 1165).

## 4.2 Publikacje wchodzące w skład osiągnięcia

[A1] Cichoń, J., Krzywiecki, Ł., Kutyłowski, M., Wlaź, P.: *Anonymous Distribution of Encryption Keys in Cellular Broadcast Systems*, MADNES. Volume 4074 of Lecture Notes in Computer Science, Springer (2005) 96–109.

[A2] Krzywiecki, Ł., Kutyłowski, M., Nikodem, M.: *General anonymous key broadcasting via Lagrangian interpolation*. IET Information Security **2** (3) (2008) 79–84

[A3] Krzywiecki, Ł., Kubiak, P., Kutyłowski, M.: *A Revocation Scheme Preserving Privacy*, Inscrypt. Volume 4318 of Lecture Notes in Computer Science, Springer (2006) 130–143.

[A4] Krzywiecki, Ł., Kutyłowski, M.: *Coalition Resistant Anonymous Broadcast Encryption Scheme Based on PUF*, TRUST. Volume 6740 of Lecture Notes in Computer Science, Springer (2011) 48–62

[A5] Krzywiecki, Ł.: *Schnorr-Like Identification Scheme Resistant to Malicious Subliminal Setting of Ephemeral Secret*, SecITC. Volume 10006 of Lecture Notes in Computer Science. (2016) 137–148.

[A6] Krzywiecki, Ł., Kutyłowski, M.: *Security of Okamoto Identification Scheme: a Defense against Ephemeral Key Leakage and Setup*, SCC@AsiaCCS, ACM (2017) 43–50.

[A7] Krzywiecki, Ł., Kutyłowski, M., Wszoła, M.: *Brief Announcement: Anonymous Credentials Secure to Ephemeral Leakage*, 2017, CSCML. Volume 10332 of Lecture Notes in Computer Science., Springer (2017) 96–98.

[A8] Krzywiecki, Ł., Słowik, M.: *Strongly Deniable Identification Schemes Immune to Prover's and Verifier's Ephemeral Leakage*, SECITC. Volume 10543 of Lecture Notes in Computer Science, Springer (2017) 115–128.

[A9] Kutyłowski, M., Krzywiecki, Ł., Kubiak, P., Koza, M.: *Restricted Identification Scheme and Diffie-Hellman Linking Problem*, INTRUST. Volume 7222 of Lecture Notes in Computer Science, Springer (2011) 221–238.

[A10] Hanzlik, L., Krzywiecki, Ł., Kutyłowski, M.: *Simplified PACE|AA protocol*, ISPEC. Volume 7863 of Lecture Notes in Computer Science, Springer (2013) 218–232.

[A11] Hanzlik, L., Kluczniak, K., Krzywiecki, Ł., Kutyłowski, M.: *Mutual Chip Authentication*, TrustCom/ISPA/IUCC, IEEE Computer Society (2013) 1683–1689.

[A12] Hanzlik, L., Kluczniak, K., Kutyłowski, M., Krzywiecki, Ł.: *Mutual Restricted Identification*, EuroPKI. Volume 8341 of Lecture Notes in Computer Science, Springer (2013) 119–133.

[A13] Krzywiecki, Ł., Kluczniak, K., Kozieł, P., Panwar, N.: *Privacy-oriented dependency via deniable SIGMA protocol*, Computers & Security **79** (2018) 53–67

[A14] Krzywiecki, Ł., Wlisłocki, T.: *Deniable key establishment resistance against eKCI attacks*, Security and Communication Networks **2017** (2017) 7810352:1–7810352:13.

[A15] Dolev, S., Krzywiecki, Ł., Panwar, N., Segal, M.: *Vehicle authentication via monolithically certified public key and attributes*, Wireless Networks **22**(3) (2016) 879–896

[A16] Dolev, S., Krzywiecki, Ł., Panwar, N., Segal, M.: *Dynamic attribute based vehicle authentication*, Wireless Networks **23**(4) (2017) 1045–1062

[A17] Dolev, S., Łukasz Krzywiecki, Panwar, N., Segal, M.: *Optical PUF for non-forwardable vehicle authentication*, Computer Communications **93** (2016) 52–67.

## 4.3 Omówienie cyklu prac

### 4.3.1 Wstęp

Niniejsza część autoreferatu stanowi opis uzyskanych przeze mnie wyników, dotyczących zapewniania prywatności użytkownikom systemów kryptograficznych, w ramach których dwie strony, mające dostęp do niezabezpieczonego kanału komunikacyjnego, wykonują pewien protokół w celu ustalenia symetrycznego klucza (tzw. klucz sesyjny), za pomocą którego szyfrowana będzie dalsza komunikacja pomiędzy nimi. Uzyskane przeze mnie wyniki można podzielić na dwa nurty ze względu na rodzaj wykorzystywanego w protokołach kanału transmisyjnego.

Pierwsza część dotyczy protokołów wykorzystujących kanał rozgłoszeniowy, w którym jeden nadawca rozgłasza szyfrowane komunikaty do wybranej grupy odbiorców. Kiedy skład grupy odbiorców ulega zmianie, następuje również zmiana klucza szyfrowania dla tej grupy. Protokoły, zaproponowane przeze mnie w tej części, zapewniają efektywną złożoność komunikacyjną oraz anonimowość użytkowników, zarówno tych uprawnionych do uzyskania nowego klucza, jak i tych, którzy z grupy zostają usunięci. W przyjętym modelu bezpieczeństwa adwersarz nasłuchujący w dostępnym kanale komunikacyjnym nie poznaje ustalanego klucza dla nowej grupy odbiorców, ani nie poznaje tożsamości odbiorców (w szczególności tożsamości osób usuniętych z systemu).

Druga część uzyskanych wyników dotyczy protokołów dla dwustronnych kanałów transmisyjnych. Dwie strony, nawiązujące bezpieczną komunikację pomiędzy sobą, identyfikują swoją tożsamość i ustalają wspólny klucz, który pozostaje nieznany dla postronnych obserwatorów. W ramach tej części wyróżniłem schematy niezbędne dla uwierzytelnionej wymiany klucza w różnych scenariuszach: schematy identyfikacji tożsamości, schematy uwierzytelniania posiadanych atrybutów, schematy uwierzytelniania tych samych osób w różnych sektorach branżowych. Zwróćmy uwagę na to, że konstrukcja schematów do ustalania wspólnego klucza szyfrowania może być modułowa, a w fazie uwierzytelniania może opierać się na kombinacji wcześniej wspomnianych schematów identyfikacji. Podstawowym wymogiem bezpieczeństwa dla tych schematów jest tajność ustalonego klucza sesyjnego - pozostaje on nieznany dla adwersarza obserwującego kanał transmisyjny.

Dodatkowym wymaganiem jest ochrona prywatności użytkowników, rozumiana jako ukrycie ich tożsamości wobec osób nieuprawnionych bądź jako możliwość zaprzeczenia udziału w wykonaniu protokołu.

Zapewnienie prywatności użytkowników uczestniczących w proponowanych protokołach w sposób wyróżniający charakteryzuje omawiane schematy kryptograficzne i w porównaniu do schematów omawianych we wcześniejszej literaturze jest nowym podejściem do omawianego problemu ustalania kluczy. Prywatność jest tutaj rozumiana dwojako. W schematach rozgłoszeniowych oznacza anonimowość użytkowników uprawnionych i wykluczanych, których tożsamości pozostają znane jedynie nadawcy. W schematach dwustronnych prywatność oznacza bądź tajność tożsamości użytkowników, bądź zaprzeczalność uczestnictwa. W pierwszym przypadku adwersarz obserwujący wykonywany protokół, lub uzyskujący dostęp transkryptu wymienianych wiadomości, nie poznaje tożsamości użytkowników nawet w sposób pośredni, taki jak przyporządkowanie do unikalnego klucza publicznego. W drugim przypadku identyfikatory te mogą być znane adwersarzowi, jednak ich właściciele mają możliwość zanegowania swojego udziału w protokole poprzez wskazanie procedury umożliwiającej zasymulowanie jego wykonania przez innego użytkownika.

W końcowej części dotyczącej uwierzytelnionego kluczy sesyjnych dla dwustronnych kanałów transmisyjnych omawiam wyniki uzyskane dla specyficznych protokołów wykorzystywanych w komunikacji pomiędzy pojazdami. Protokoły z tej grupy mają konstrukcję dwuwarstwową. Pierwsza warstwa, realizowana zazwyczaj w kanale radiowym, odpowiada za funkcjonalność uwierzytelnionego ustalania klucza, np. metodami opisanymi powyżej. Natomiast druga warstwa, realizowana w kanale optycznym, jest odpowiedzialna za dodatkową identyfikację wizualnych atrybutów pojazdów. Wiązanie pomiędzy warstwami realizowane jest za pomocą certyfikatów łączących klucz publiczny protokołów warstwy radiowej z ustalonymi atrybutami wyglądu weryfikowanymi optycznie.

### 4.3.2 Struktura autorefertu

W rozdziale 4.3.3 opisuję prace [A1, A2, A3, A4] dotyczące kodowanego rozgłaszania klucza sesyjnego od jednego nadawcy do wielu uprawnianych użytkowników we współdzielonym kanale komunikacyjnym. Główną techniką wykorzystywaną w tych pracach jest schemat dzielenia tajemnic oparty na interpolacji w wykładniku pewnego tajnego wielomianu należącego do nadawcy. Nowy klucz sesyjny rozsyłany do uprawnionych użytkowników kodowany jest za pomocą zbioru interpolacyjnego zawierającego udziały użytkowników wykluczanych. Głównym rezultatem, uzyskanym dla tych schematów, jest takie przetwarzanie udziałów, aby na podstawie rozgłaszanych wartości nie można było zidentyfikować tożsamości użytkowników wykluczonych z odbioru. W pracy [A1] zaproponowałem pierwszy anonimowy schemat kodowanego rozgłaszania oparty na interpolacji Lagrange'a dla systemów, w których liczba użytkowników wykluczanych zmienia się dynamicznie, a jej maksymalna wartość nie jest znana *a priori*. Schemat ten rozszerzyłem w pracy [A2], uzyskując dwukrotnie mniejszą złożoność komunikacyjną przy tych samych założeniach. W pracy [A3] zaproponowałem anonimowy schemat kodowanego rozgłaszania dla systemów, w których maksymalna liczba wykluczanych użytkowników w sesji jest znana, i znacznie mniejsza niż liczba użytkowników pozostających w systemie. Z kolei schemat zaproponowany w pracy [A4] jest pierwszym anonimowym schematem kodowanego rozgłaszania klucza, którego bezpieczeństwo oparte jest na funkcjach fizycznie nieklonowalnych.

Pozostałe prace dotyczą dwustronnego kanału komunikacyjnego. W pracach tych zaproponowano protokoły identyfikacji i ustalania kluczy sesyjnych, w których użytkownicy posługujący się certyfikowanymi kluczami asymetrycznymi, identyfikują wzajemnie swoją tożsamość, oraz ustalają symetryczne klucze sesyjne służące do dalszego szyfrowania kanału komunikacyjnego. W kontekście modułowej budowy protokołów kryptograficznych, protokoły identyfikacji mogą być wykonywane jako podprocedury ustalania tożsamości dla bardziej złożonych protokołów uwierzytelnionego ustalania klucza pomiedzy dwoma stronami.

W rozdziale 4.3.4 omawiam prace [A5, A6, A8, A7, A9], w których zaproponowałem zaprzeczalne protokoły identyfikacji i uwierzytelniania atrybutów użytkowników. Dla tych protokołów, w pracach [A5, A6, A8, A7], zaproponowałem nowy silniejszy model bezpieczeństwa, w którym adwersarz pomimo adaptatywnego ustalania w fazie uczenia wartości losowych na urządzeniach użytkowników uwierzytelnianych, nie poznaje długoterminowych kluczy tajnych, ani nie jest w stanie wykonać fałszywej identyfikacji (”atak podszycia”), udając właściwego użytkownika bez znajomości jego klucza tajnego. Schematy z prac [A5, A6] są modyfikacjami klasycznych schematów identyfikacji Schnorra i Okamoto, które osiągają bezpieczeństwo w proponowanym modelu. Prywatność użytkowników w tych schematach zapewniana jest poprzez możliwość zaprzeczenia, wobec zewnętrznego obserwatora, udziału użytkownika w protokole przeprowadzonym wraz z uczciwym weryfikatorem. Zaprzeczenie takie nie jest jedynie deklaratywne, ale oparte o silne argumenty natury kryptograficznej. Dodatkowo w pracy [A8] zaproponowałem metody przeciwdziałania atakom przeprowadzanym przez nieuczciwego weryfikatora, którego celem jest utworzenie niezaprzeczalnego transkryptu protokołu za pomocą transformacji Fiata-Shamira. W pracy tej, model bezpieczeństwa rozrzerzono na przypadek, w którym adwersarz przeprowadzający atak ”podszycia” poznaje wartości losowe wykorzystywane przez weryfikatora. W pracy [A9] zaproponowano protokół identyfikacji użytkownika w wielu sektorach, za pomocą jednego klucza prywatnego w taki sposób, że niemożliwe jest przyporządkowanie wykonań protokołu w różnych sektorach do jednego użytkownika - właściciela jednego klucza prywatnego. Dzięki temu identyfikacja użytkownika w jednym sektorze nie może być powiązana z jego identyfikacją w innym sektorze.

W rozdziale 4.3.5 omawiam prace [A10, A11, A12, A13, A14, A15, A16, A17] dotyczące nowych protokołów uwierzytelnionej wymiany kluczy sesyjnych. W pracach [A10, A11, A12] zaproponowano nowe, wprowadzające zaprzeczalność i anonimowość, protokoły wymiany klucza dla dokumentów elektronicznych. Schemat z pracy [A10] jest ulepszoną, odporną na wyciek wartości efemerycznych, wersją niemieckiego protokołu PACE|AA [1] zgodnego ze standardem International Civil Aviation Organization [2] do maszynowej autoryzacji dokumentów podróżnych (obecnie nazwa używana w standardzie ICAO to PACE-CAM). Z kolei, w pracy [A11] zaproponowano protokół do uwierzytelniania karty mikroprocesorowej i terminala, będący alternatywą dla protokołów zalecanych przez niemiecki urząd do spraw bezpieczeństwa informatycznego (Bundesamt für Sicherheit in der Informationstechnik - w skr. BSI) dla elektronicznych dokumentów tożsamości w standardzie [3]. Dodatkowo dla tego standardu, w pracy [A12] zaproponowano zoptymalizowaną zaprzeczalną wersję protokołu uwierzytelnionego ustalania klucza sesyjnego, w różnych sektorach aktywności użytkownika. Prace [A13, A14] dotyczą ogólniejszych protokołów stosowanych w systemach internetowych. Praca [A13], będąca rozszerzoną wersją mojej pracy konferencyjnej [4], zawiera propozycję modyfikacji protokołu wymiany kluczy sesyjnych SIGMA z [5], zapewniającą możliwość zaprzeczenia udziału za pomocą wykorzystania podpisów pierścieniowych oraz zwiększającą bezpieczeństwo klucza sesyjnego w razie kompromitacji kluczy efemerycznych. W pracy [A14] zaproponowałem rozszerzenie protokołu HMQV [6] zapewniające jednocześnie zaprzeczalność oraz odporność na ataki eKCI [7]. W pracach [A15, A16, A17] zaproponowano protokoły identyfikacji i ustalania kluczy sesyjnych w systemach komunikacji pojazdów. W przeciwieństwie do poprzednich prac, protokoły omawiane tutaj wiążą strony, uwierzytelniane asymetrycznym kluczem publicznym, z pojazdem o certyfikowanym wyglądzie i atrybutach fizycznych identyfikowanych w kanale optycznym. W pracy [A15] atrybytami tymi są ustalone statyczne właściwości pojazdu takie jak kolor, marka, model. W pracy [A16] dodatkowo rozważano identyfikowanie atrybutów dynamicznych: położenie, prędkość. Z kolei w pracy [A17] zaproponowałem protokół identyfikacji pojazdów, wykorzystujący funkcje fizycznie nieklonowalne, a odporny na ataki typu "man-in-the-middle"polegające na przekierowaniu kanału komunikacyjnego do zdalnego adwersarza. Wyniki otrzymane w pracach [A15, A16, A17] stanowią podstawę uzyskania przez autorów patentu w USA [8].

**Oznaczenia, założenia i wykorzystywane techniki** Niech $x_1, \dots, x_n \leftarrow_\$ X$ oznacza, że każde $x_i$ jest wybierane losowo z rozkładem jednostajnym ze zbioru $X$. Dodatkowo, dla uproszczenia stylu autoreferatu, przyjmujemy, że sformułowanie “wybieranie losowe” oznacza losowanie z rozkładem jednostajnym z odpowiedniego zbioru wynikającego z kontekstu. Niech $\mathcal{G}(\lambda)$, będzie algorytmem przyjmującym parametr $\lambda$ i generującym przestrzeń obliczeń opisany krotką $\mathbb{G} = (p, q, g, G)$, taką że $p, q$ są liczbami pierwszymi oraz $q|p-1$. $\mathbb{Z}_p^*$ jest grupą multiplikatywną modulo $p$. Element $g \in \mathbb{Z}_p^*$ jest rzędu rzędu $q$. Przez $G$ będziemy oznaczać podgrupę grupy $\mathbb{Z}_p^*$ generowaną przez $g$. Parametr $\lambda$ jest nazywany parametrem bezpieczeństwa, który definiuje wielkość liczb $p, q$. Dla grupy $G$ można przyjmować założenia o trudności następujących problemów:
1) *Problem dyskretnego logarytmu* (DL): Dla każdego algorytmu $\mathcal{A}_{\mathsf{DL}}$ działającego w czasie wielomianowym zachodzi: $\Pr[\mathcal{A}_{\mathsf{DL}}(\mathbb{G}, g^x) = x \mid \mathbb{G} \leftarrow_\$ \mathcal{G}(\lambda), x \leftarrow_\$ \mathbb{Z}_q^*] \leq \epsilon_{\mathsf{DL}}(\lambda)$, gdzie $\epsilon_{\mathsf{DL}}(\lambda)$ jest małą “zaniedbywalną” wielkością.
2) *obliczeniowy problem Diffiego-Hellmana* (CDH): Dla każdego algorytmu $\mathcal{A}_{\mathsf{CDH}}$ działającego w czasie wielomianowym: $\Pr[\mathcal{A}_{\mathsf{CDH}}(\mathbb{G}, g^x, g^y) = g^{xy} \mid \mathbb{G} \leftarrow_\$ \mathcal{G}(\lambda), x \leftarrow_\$ \mathbb{Z}_q^*, y \leftarrow_\$ \mathbb{Z}_q^*] \leq \epsilon_{\mathsf{CDH}}(\lambda)$, gdzie $\epsilon_{\mathsf{CDH}}(\lambda)$ jest zaniedbywalne.
3) *Decyzyjny problem Diffiego-Hellmana* (DDH): Niech $\mathbb{G} \leftarrow_\$ \mathcal{G}(\lambda)$, $x \leftarrow_\$ \mathbb{Z}_q^*$ , $y \leftarrow_\$ \mathbb{Z}_q^*$, $z \leftarrow_\$ \mathbb{Z}_q^*$, $D_0 = (\mathbb{G}, g^x, g^y, g^{xy})$, $D_0 = (\mathbb{G}, g^x, g^y, g^z)$. Dla każdego algorytmu $\mathcal{A}_{\mathsf{DDH}}$ działającego w czasie wielomianowym: $|\Pr[\mathcal{A}_{\mathsf{DDH}}(D_0) = 0] - \Pr[\mathcal{A}(D_1) = 0]| \leq \epsilon_{\mathsf{DDH}}(\lambda)$, gdzie $\epsilon_{\mathsf{DDH}}(\lambda)$ jest zaniedbywalne.
4) *obliczeniowy problem Diffiego-Hellmana z wyrocznią decyzyjną* (GDH): Niech $\mathcal{O}_{\mathsf{DDH}}$ oznacza algorytm (nazywany *wyrocznią decyzyjną Diffiego-Hellmana*) działający w czasie wielomia-

nowym, który dla $\mathbb{G} \leftarrow_\$ \mathcal{G}(\lambda), x \in \mathbb{Z}_q^*, y \in \mathbb{Z}_q^*, z \in \mathbb{Z}_q^*$ oblicza $\mathcal{O}_{\mathsf{DDH}}(\mathbb{G}, g^x, g^y, g^z) = 1$ wtedy i tylko wtedy, gdy $z = xy \mod q$. Dla każdego probabilistycznego algorytmu działającego w czasie wielomianowym $\mathcal{A}_{\mathsf{GDH}}^{\mathcal{O}_{\mathsf{DDH}}}$ posiadającego dostęp do wyroczni decyzyjnej Diffiego-Hellmana $\mathcal{O}_{\mathsf{DDH}}$ zachodzi: $\Pr[\mathcal{A}_{\mathsf{GDH}}^{\mathcal{O}_{\mathsf{DDH}}}(\mathbb{G}, g^x, g^y) = g^{xy} \mid \mathbb{G} \leftarrow_\$ \mathcal{G}(\lambda), x \leftarrow_\$ \mathbb{Z}_q^*, y \leftarrow_\$ \mathbb{Z}_q^*] \leq \epsilon_{\mathsf{GDH}}(\lambda)$, gdzie $\epsilon_{\mathsf{CDH}}(\lambda)$ jest zaniedbywalne.

Parametr bezpieczeństwa $\lambda$, który definiuje wielkość liczb $p, q$ jest dobierany tak aby liczność $G$ była odpowiedni duża, tzn. aby powyższe problemy nie były w praktyce rozwiązywalne za pomocą przeszukiwań wyczerpujących. Upraszczając nieco, w grupach, w których przyjęto założenie DL można konstruować kryptosystemy asymetryczne, w których kluczowi prywatnemu $\mathsf{sk} \in \mathbb{Z}_q^*$ odpowiada klucz publiczny $\mathsf{pk} = g^{\mathsf{sk}}$. Analogicznie w przy założeniu CDH buduje się protokoły ustalania kluczy sesyjnych przez dwie strony wymieniające się wzajemnie wartościami $g^x, g^y$ i wyliczającymi niezależnie po każdej stronie wartość $g^{xy}$.

Niech $G$ oznacza grupę cykliczną rzędu pierwszego $q$, oraz $G_T$ będzie inną cykliczną grupą rzędu $q$. Wtedy $\hat{e} : G \times G \to G_T$ jest odwzorowanie dwuliniowym, gdy następujące warunki są spełnione :

1) *Dwuliniowość*: $\forall a, b \in \mathbb{Z}_q^*, \forall g \in G$: $\hat{e}(g^a, g^b) = \hat{e}(g, g)^{ab}$.
2) *Niedegnerowalność*: $\hat{e}(g, g) \neq 1$.
3) *Obliczalność*: $\hat{e}$ jest efektywnie obliczalne.

Funkcja $\hat{e}$ jest wykorzystywana do tworzenia kryptosystemów opartych na problemie GDH, w których zakłada się trudność CDH natomiast $\hat{e}$ pełni rolę wyroczni $\mathcal{O}_{\mathsf{DDH}}$.

Niech $L \in \mathbb{Z}_p[x]$ będzie wielomianem stopnia $k < p$. Niech $A = \langle (x_0, L(x_0)), \ldots, (x_k, L(x_k)) \rangle$ będzie zbiorem par liczb takich, że $x_i \neq x_j$ dla $i \neq j$. Przez

$$LI_A(x) \ = \sum_{i=0}^{k} \left( L(x_i) \prod_{j=0, j\neq i}^{k} \left( \frac{x - x_j}{x_i - x_j} \right) \right) \tag{1}$$

oznaczamy interpolację Lagrange'a wielomianu $L$.

Niech $(p, q, g, G) \leftarrow \mathcal{G}(\lambda)$, będzie krotką taką że $p, q$ są liczbami pierwszymi oraz $q | p - 1$, a $G$ oznacza podgrupę grupy $\mathbb{Z}_p^*$ generowaną przez $g$. Niech $(x_0, \ldots, x_k) \in \mathbb{Z}_q^k$, $r \in \mathbb{Z}_q$, oraz $L \in \mathbb{Z}_q[x]$ stopnia $k < q$. Ciąg $A' = \langle (x_0, g^{rL(x_0)}), \ldots, (x_k, g^{rL(x_k)}) \rangle$, gdzie $x_i \neq x_j$ dla $i \neq j$, nazywamy zbiorem punktów interpolacyjnych (gdzie w wykładnikach wykonuje się działania z ciała $\mathbb{Z}_q$. Wtedy funkcję $LI_{EXP} : \mathbb{Z}_q \to G$ określoną wzorem:

$$LI_{EXP,A'}(x) \ = \ \prod_{i=0}^{k} g^{rL(x_i) \cdot \prod_{j=0, j\neq i}^{k} \left( \frac{x - x_j}{x_i - x_j} \right)}.$$

nazywamy interpolacją Lagrange'a w wykładniku. Zauważmy, że $(\forall_{a \in \mathbb{Z}_q})(g^{a+q} = g^a)$, z czego wynika, że $LI_{EXP,A'}(a) = g^{rL(a)}$.

### 4.3.3 Ustalanie kluczy w systemach rozgłoszeniowych

Kodowane rozgłaszanie (ang. Broadcast Encryption (BE)) jest ~~z~~ schematem kryptograficznym, w którym *nadawca* wysyła wiadomości do uprawnionych użytkowników poprzez kanał rozgłoszeniowy. Przykładem wykorzystanie takiego schematu może być system płatnej telewizji odbieranej jedynie przez subskrybentów opłacających abonament, np. miesięczny. W dalszej części autoreferatu okres nadawania, w którym grupa użytkowników uprawnionych jest ustalona nazywamy sesją. Dodatkowo oznaczmy przez $N$ liczbę wszystkich zarejestrowanych użytkowników, oraz przez $z$ liczbę użytkowników nieuprawnionych. Jeśli grupa $N - z$ uprawnionych odbiorców ulega zmianie to następuje przejście do nowej sesji. Zakłada się, że wykorzystywany do nadawania kanał rozgłoszeniowy jest dostępny dla każdego nasłuchującego. W związku z tym nadawca koduje wiadomości w rozgłaszanych danych tak, aby spełnić następujące wymagania: a) nieuprawnieni użytkownicy nie mogą odkodować wiadomości z rozgłaszanych danych; b) każdy uprawniony użytkownik powinien odkodować wiadomości z rozgłaszanych danych; c) system powinien umożliwiać, w sposób efektywny, kodowanie nowych wiadomości do zmieniającego się zbioru uprawnionych użytkowników w każdej kolejnej sesji.

Intuicyjnym sposobem osiągnięcia tych właściwości jest rozsyłanie danych, zaszyfrowanych za pomocą wybranego szybkiego schematu symetrycznego, i dostarczenie klucza deszyfrującego - zwanego *sesyjnym* - każdemu uprawnionemu użytkownikowi. Problem rozsyłania kluczy sesyjnych dla systemów szyfrowanego rozgłaszania postawiono w pracy [9]. W przypadku istnienia Infrastruktury Klucza Publicznego (IKP), klucz sesyjny można szyfrować odpowiednimi kluczami publicznymi poszczególnych uprawnionych użytkowników. Wtedy złożoność komunikacyjna protokołu jest proporcjonalna do liczby użytkowników uprawnionych, którzy nie są wykluczeni z rozgłaszania. Narzut ten jest wysoki, jeśli ze zbioru uprawnionych osób wykluczamy jedynie niewielką liczbę użytkowników $z$, bowiem wiąże się z wysyłaniem dodatkowych wiadomości do wszystkich $N - z$ użytkowników, poza wykluczonymi. W tych przypadkach poszukuje się schematów, w których narzut komunikacyjny, związany z wysyłaniem nowego klucza sesyjnego do uprawnionych użytkowników, byłby proporcjonalny do małej liczby użytkowych wykluczonych. Zaproponowano wiele protokołów, które rozwiązują ten problem np. [10, 11, 12, 13, 14, 15, 16, 17]. W przedstawianych w autoreferacie pracach koncentrujemy się na protokołach BE opartych na schemacie *dzielenia tajemnic* Shamira i interpolacji Lagrange'a [13, 14, 15, 16]. Dla tych systemów zakładamy istnienie infrastruktury sprzętowej, w której użytkownik $u$ podczas procesu rejestracji dostaje urządzenie (tzw. dekoder), zawierające tajny klucz przyporządkowany przez nadawcę. Dekoder pobiera dane z kanału rozgłoszeniowego i, jeśli użytkownik jest uprawniony, dekoduje wiadomości, wykonując niezbędne obliczenia z wykorzystaniem tajnego klucza.

**Zmiana zbioru odbiorców za pomocą interpolacji Lagrange'a** Interpolacja Lagrange'a jest jedną z podstawowych technik wykorzystywanych przy projektowaniu systemów rozgłaszania, odpowiednich dla wykluczania małej liczby osób. Systemy te buduje się w oparciu pewien ustalony przez nadawcę tajny wielomian, którego stopień definiuje maksymalną liczbę użytkowników wykluczanych w pojedynczej sesji.

Rozgłaszający wybiera sekretny wielomian $L$ i dostarcza zarejestrowanemu użytkownikowi $u$ udziały w postaci pary liczb $(x_u, f(L(x_u)))$, gdzie $x_u$ jest losową wartością z dziedziny wielomianu $L$, a $f$ jest funkcją, która ukrywa postać wielomianu przed użytkownikiem, uniemożliwiając mu poznanie tajnych współczynników, nawet jeśli ten sam wielomian jest używany w wielu następujących po sobie sesjach. Przykładem takiej funkcji jest $f(L(x_u)) = g^{L(x_u)}$, gdzie operacje są wykonywane w grupie $G$ generowanej przez $g$, w której obliczanie dyskretnego logarytmu jest problemem trudnym obliczeniowo.

Nowy klucz sesyjny kodowany jest jako $f(rL(x_0))$, dla wybranego $x_0$ i losowej wartości $r$, która dodatkowo maskuje wielomian $L$. Nadawca tworzy nagłówek z danymi, zawierający między innymi argument $x_0$, oraz $z$ par liczb postaci $(x, f(rL(x)))$. Wśród nich: a) znajdują się udziały wszystkich wykluczanych użytkowników; b) nie ma udziałów użytkowników uprawnionych. Zauważmy, że pary te tworzą niekompletny zbiór interpolacyjny - brakuje dokładnie jednej pary do przeprowadzenia interpolacji Lagrange'a. Po otrzymaniu nagłówka uprawniony użytkownik $u$ wykonuje następujące czynności: a) przygotowuje swój udział $(x_u, f(rL(x_u)))$; b) wykorzystuje swój udział i pary z nagłówka do obliczenia $f(rL(x_0))$. Zauważmy, że żaden wykluczony użytkownik nie może przeprowadzić interpolacji Lagrange'a - jego własny udział znajduje się już w nagłówku - dysponuje więc niepełnym zbiorem iterpolacyjnym.

Podstawowymi zagrożeniami dla schematów budowanych w ten sposób są

- Duplikacja urządzeń: Jeśli adwersarzowi udałoby się wydobyć materiał kryptograficzny z urządzenia służącego do dekodowania, to mógłby on zbudować kopię takiego urządzenia, a następnie deszyfrować rozgłaszane dane na każdej kopii.

- Koalicje adwersarzy: Wykonywanie obliczeń w wykładniku, oraz maskowanie wielomianu $L$ w kolejnych sesjach nowymi losowymi wartościami $r$, uniemożliwia pojedynczym adwersarzom poznanie tajnego wielomianu $L$. Pomimo tego istnieje niebezpieczeństwo, że koalicja $z + 1$ adwersarzy, dzieląc się swoimi udziałami, może próbować zrekonstruować tajny wielomian $L$ nadawcy, a następnie zbudować urządzenie, którego nie dałoby się wykluczyć.

- Zagrożenie prywatności: W typowych schematach rozgłaszania z wykluczaniem opartych na technikach wielomianowych możliwe jest stwierdzenie czy ten sam dekoder został wykluczany w wielu sesjach, co może posłużyć analizie zachowań poszczególnych użytkowników i zagrażać ich prywatności.

We wszystkich wcześniejszych pracach opartych na interpolacji Lagrange'a długość nagłówka jest proporcjonalna do liczby użytkowników wykluczanych $z$, co jest niewątpliwie zaletą obniżająca narzut komunikacyjny protokołu. Jednak występowanie w nagłówku udziałów użytkowników wykluczanych można potraktować jako wadę schematu. Znajomość tych udziałów może posłużyć adwersarzowi do naruszenia prywatności użytkowników.

**System BE oparty na dobieraniu przedziałami** W pracy [A1] zakładamy, że w systemie jest $n$ użytkowników oraz, że do $m$-elementowej $(m \leq n)$ podgrupy użytkowników należy wysłać nowy klucz sesyjny. Ustalamy dwie bezpieczne kryptograficznie funkcje skrótu $H, H' : \{0,1\}^* \rightarrow \mathbb{Z}_p = \{0, \ldots, p-1\}$, gdzie $p$ jest ustaloną dużą liczbą pierwszą $(p \gg m)$. Dla wygody zapisu przyjmujemy notację, w której przecinek pomiędzy argumentami funkcji $H, H'$, oznacza konkatenację tych argumentów. Niech $A_1$, $A_2$ …, $A_m$ będą identyfikatorami użytkowników, którzy mają otrzymać nowy klucz $K$. Nadawca będzie kodował klucz w pseudolosowym wielomianie zdeterminowanym udziałami uprawnionych użytkowników, ale podczas nadawania będzie przesyłał inne punkty należące do tego wielomianu, aby ukryć dla jakich użytkowników dany wielomian powstał. Nadawca wybiera losową wartość $q$. Następnie nadawca oblicza $u_i := H(q, s(A_i))$ i $x_i := H'(q, s(A_i))$ dla $i = 1 \ldots, m$, gdzie $s(A_i)$ oznacza sekret przyporządkowany użytkownikowi $A_i$. Ustalmy, że $x_i > m$. Nadawca wyznacza wielomian $f \in \mathbb{Z}_p[x]$ stopnia $m$ taki, że $f(x_i) = u_i$ dla $i = 1, \ldots, m$.

W procedurze kodowania klucza nadawca wylicza i nadaje nagłówek $\langle q, f(1), \ldots, f(m)\rangle$. Użytkownik uprawniony $A_i$ może zrekonstruować $K$ z wartości wielomianu $f$ w $m + 1$ różnych punktach za pomocą zbioru interpolacyjnego $\langle(1, f(1)), \ldots, (m, f(m)), (x_i, u_i)\rangle$. Schemat zapewnia anonimowość użytkowników $A_1, A_2, \ldots, A_m$, gdyż określenie wartości $x_i, u_i$, dla ustalonego uprawnionego użytkownika $A_i$, wymagałoby znalezienia takiego $s$, że $x_i = H(q, s)$ i $u_i = H'(q, s)$, oraz $f(x_i) = u_i$. W sytuacji, gdy liczba użytkowników uprawnionych $m'$ jest mniejsza niż $m$, to nadawca ustala $m - m'$ wartości $f$ w punktach $1, \ldots, m - m'$ sposób losowy.

Pewną wadą powyższego rozwiązania jest koszt obliczeń po stronie użytkownika. Po otrzymaniu wartości $f(1), \ldots, f(m)$ oraz $q$, użytkownik musi wykonać $\Omega(m)$ operacji arytmetycznych. Dlatego w dalszej części tego schematu zaproponowano metodę dzielenia zbioru uprawnionych na przedziały zwane urnami i kodowanie klucza dla każdej urny oddzielnie. Zakładamy, że $\mathcal{U} = \{A_1, \ldots, A_m\}$, gdzie $m \leq n$, jest zbiorem użytkowników uprawnionych do odkodowania $K$. W trakcie kodowania klucza każdy użytkownik jest przydzielony pseudolosowo do jednej spośród $B$ urn. Dla każdej urny klucz sesyjny jest kodowany niezależnie metodą przedstawioną powyżej w poprzednim akapicie. Takie rozwiązanie zapewnia następujące własności: a) każdy użytkownik wie, w której urnie powinien odkodowywać klucz sesji; b) użytkownicy nie pozyskują danych o innych użytkownikach; c) w każdej urnie klucz kodowany jest dla podobnej liczby użytkowników

Aby pseudolosowe przyporządkowanie użytkowników do urn było równomierne wykorzystujemy technikę tzw. "zbalansowanej alokacji kul do urn" (ang. balls-into-bins balanced allocation) [18, 19, 20, 21, 22]. Przypomnijmy, że jeśli alokacja $n$ kul odbywa się sekwencyjnie i wybierana jest mniej wypełniona urna z $d$ losowo wybranych spośród wszystkich $B$ urn, to maksymalnie wypełnienie urn wynosi $n/B + \log \log B / \ln d + o(1)$ z dużym prawdopodobieństwem [19, 21]. Dalsza część protokołu opiera się na procedurze *left*$[d]$ z pracy [22]. Procedura ta *left*$[d]$ rozbija zbiór $n$ urn na $d$ podzbiorów o rozmiarze $n/d$. Przyjmujemy, że urny te ustawione sa w porządku od 1 do $n$, gdzie pierwszy, skrajnie lewy podzbiór, zawiera urny od 1 do $d$, itd. Procedura *left*$[d]$ wykonywana jest sekwencyjnie: dla kolejnej kuli z każdego z $d$ podzbiorów wybierana jest losowa urna - w sumie $d$ urn. Następnie kula jest wstawiana do tej z wybranych urn, która jest najmniej wypełniona. W przypadku, gdy kilka urn ma tę samą minimalną liczbę kul, wybierana jest skrajnie lewa urna spośród nich.

W protokole nadawca przyporządkowuje użytkownika $A$ do najmniej wypełnionej urny spośród $d$ losowo wybranych. Wybór jest pseudolosowy i zależny od klucza, który nadawca dzieli z użytkownikiem - a zatem możliwy do wyliczenia jedynie przez nich. Jednak tylko nadawca wie, która urna jest najmniej wypełniona. Klucz sesyjny $K$ może być odkodowany jedynie z wiadomości przyporządkowanych urnom, zatem użytkownik w najgorszym przypadku próbuje odkodować klucz dla każdej z urn do których może być przyporządkowany, czyli $d$-krotnie. Natomiast każda urna jest przyporządkowana do transmisji nowego klucza dla pewnej grupy użytkowników przypisanych do niej. Nadawca decyduje, który z użytkowników przypisanych do urny, zdekoduje prawidłowo klucz sesyjny $K$. Dla każdej urny nagłówek kodujący klucz $K$ jest konstruowany metodą opisana powyżej.

**Anonimowy system BE oparty na wykluczaniu przedziałami i *dobieraniu*** W pracy [A2] zaproponowałem schemat zarządzania kluczami dla sieci rozgłoszeniowych, który jest złożeniem dwóch typów protokołów kodowanego rozgłaszania: a) wykluczania użytkowników nieuprawnionych, np. [23, 11, 13, 24], gdy liczba wykluczanych użytkowników jest mała; b) dobierania użytkowników uprawnionych (propozycja z [A1]), gdy liczba użytkowników wykluczanych jest duża. Proponowany schemat łączy te dwie techniki w taki sposób, aby informacje o tym kto jest wykluczany i kiedy, były niedostępne dla obserwujących transmisję adwersarzy, a ponadto aby złożoność komunikacyjna systemy była niezależna od zmian liczby użytkowników uprawnionych. System taki może być wykorzystywany w przypadku, gdy liczba takich zamian jest trudna do oszacowania. W zaproponowanym schemacie przeciętny narzut komunikacyjny - w przypadku kiedy dynamika zmian zbioru użytkowników uprawnionych jest duża, a maksymalna liczba użytkowników wykluczanych lub uprawnionych nie jest znana "a priori" - jest o połowę mniejszy niż w przypadku schematu z pracy [A1]). Dodatkowo, tożsamość użytkowników wykluczanych jest ukryta przed adwersarzem.

Poprzednie schematy wykluczania rozwiązywały problem narzutu komunikacyjnego, gdy liczba $k$ użytkowników wykluczanych była mała ($k \ll N/2$, gdzie $N$ oznacza liczbę wszystkich użytkowników. Wtedy długość nagłówka była ustalona i proporcjonalna do stałego parametru $k$ protokołu oznaczającego maksymalną liczbę wykluczanych. W pracy rozważamy przypadek, gdy zbiór wykluczanych może mieć liczność $k > N/2$. Protokół składa się z procedur Setup, Register, Enc1, Enc2, Decoding. Pokazujemy, w jaki sposób zaprojektować procedury Setup, i Register, tak aby nadawca mógł użyć procedury wykluczania Enc1 opartej na schematach [23, 11, 13, 24], jeśli liczba użytkowników wykluczanych jest mała, oraz procedury Enc2 opartej na schemacie [A1], jeśli liczba liczba użytkowników wykluczanych jest duża. Nagłówki generowane przez obie procedury wyglądają tak samo i uniemożliwiają obserwatorowi rozróżnienie, która procedura została wykorzystana. W szczególności procedura Decoding wykonywana w dekoderze użytkownika nie zależy od wersji procedury kodowania.

W procedurze inicjalizacji nadawca $\mathcal{B}$ ustala zbiór kubełków (przedziałów). Następnie podczas rejestracji użytkownicy zostają przydzieleni do kubełków za pomocą czterech różnych mapowań w taki sposób, że każdy użytkownik jest przydzielany tylko do dwóch kubełków. Jednak zbiór użytkowników w tym samym kubełku (nazywanych sąsiadami) zależy od użytego mapowania, a każdy użytkownik w swoich kubełkach może mieć dwa różne zbiory sąsiadów. Nadawca $\mathcal{B}$ konstruuje zbiór losowych wielomianów: jeden wielomian dla każdego wyboru kubełka i mapowania.

Podczas procedury kodowania nadawca wybiera losowo jedno mapowanie i koduje klucz sesyjny niezależnie dla każdego kubełka. Jeśli zbiór użytkowników wykluczanych w kubełku jest mniejszy niż $\frac{1}{2}$, wykorzystywana jest procedura Enc1 na podstawie wielomianów przypisanych dla tego kubełka. W przeciwnym wypadku nadawca $\mathcal{B}$ wykorzystuje procedurę Enc2 wykorzystującą zupełnie nowy, tworzony "ad-hoc" wielomian, który jest różny od wielomianów przypisanych do kubełka w tym mapowaniu. Celem mapowań jest utrudnienie wnioskowania adwersarza. Użytkownik sprawdza tylko dwa różne kubełki, w których może być dla niego zakodowany klucz sesyjny. Nawet jeśli uprawniony użytkownik zdekoduje klucz sesyjny w jednym ze swoich kubełków, to wciąż nie wie, które mapowanie zostało wykorzystane podczas kodowania, oraz jaki jest aktualnie zbiór sąsiadów użytkownika w tym kubełku. W dalszej części $\Omega$ oznacza zbiór wszystkich użytkowników, a $\Phi$ oznacza zbiór użytkowników wykluczanych. Podczas procedury Setup nadawca $\mathcal{B}$ określa zbiór identyfikatorów $\Omega \subset \mathbb{N}$ użytkowników, parametry $k, q$, oraz zbiór pseudonimów użytkowników *ID* $= \{1, \ldots, 4k\}$ (przyjmujemy, że *ID* $\subset \mathbb{Z}_p$). Nadawca ustala zbiór kubełków $B$ o liczności $|\Omega|/2k$, każdy o pojemności $2k$ oraz definiuje mapowania $\Pi_1, \ldots, \Pi_4 : \Omega \to B$. Dla każdego $i$, mapowanie $\Pi_i$ przyporządkowuje identyfikatory do kubełków tak, że każdy kubełek dostaje dokładnie $2k$ identyfikatorów. Niech $\Omega_{i,b}$ będzie zbiorem identyfikatorów kubełka $b$ przyporządkowanych mapowaniem $\Pi_i$. Mapowania są zdefiniowane tak, że dla każdego identyfikatora $u \in \Omega$, istnieją indeksy $i$ oraz $j$, takie, że lista $\Pi_1(u), \Pi_2(u), \Pi_3(u), \Pi_4(u)$ zawiera liczby $i$ i $j$, pojawiające się dwukrotnie każda. Jeśli $u \in \Omega_{i,b}$, to $i'$ oznacza indeks taki, że $u \in \Omega_{i',b}$ i $i' \neq i$. Dla każdego $\Omega_{i,b}$ nadawca $\mathcal{B}$ definiuje zbiór pseudonimów $P_{i,b} \subset$ *ID* zgodnie z rozkładem jednostajnym, tak że $|P_{i,b}| = |\Omega_{i,b}| = 2k$, a następnie przyporządkowuje pseudonimy do identyfikatorów za pomocą losowej bijekcji $f_{i,b} : \Omega_{i,b} \to P_{i,b}$. Dla każdego zbioru $\Omega_{i,b}$ nadawca $\mathcal{B}$ definiuje wielomian $L_{i,b} \in \mathbb{Z}_p[X]$ stopnia $k$ o współczynnikach losowych. Nadawca $\mathcal{B}$ wybiera losową liczbę $s$. Mapowania $\Pi_i$, $f_{i,b}$, wielomiany $L_{i,b}$, oraz liczba $s$ są sekretami nadawcy.

W procedurze Register wykonywane są następujące kroki: $\mathcal{B}$ przyporządkowuje wolny identyfikator $u$ do użytkownika. Dla każdego $i = 1, \ldots, 4$, $\mathcal{B}$ ustala kubełek $\Pi_i(u)$ zawierający $u$ oraz pseudonim $x_{u,i} = f_{i,\Pi_i(u)}(u)$. Dla każdego pseudonimu $x_{u,i}$, nadawca $\mathcal{B}$ oblicza $L_{i,\Pi_i(u)}(x_{u,i})$ oraz $\alpha_{u,i}, \beta_{u,i} \in \mathbb{Z}_p$ takie, że

$$L_{i,\Pi_i(u)}(x_{u,i}) = \alpha_{u,i} + s \cdot \beta_{u,i} \; .$$

Nadawca $\mathcal{B}$ przekazuje użytkownikowi następujące krotki:

$$(x_{u,1}, \alpha_{u,1}, \beta_{u,1}), \ldots, (x_{u,4}, \alpha_{u,4}, \beta_{u,4}),$$

stanowiące klucz sekretny $PK_u$ użytkownika $u$ w następującym formacie:

$$\begin{aligned} PK_u = \langle \quad & [\Pi_{i_1}(u), (x_{u,i_1}, \alpha_{u,i_1}, \beta_{u,i_1}), (x_{u,i'_1}, \alpha_{u,i'_1}, \beta_{u,i'_1})], \\ & [\Pi_{i_2}(u), (x_{u,i_2}, \alpha_{u,i_2}, \beta_{u,i_2}), (x_{u,i'_2}, \alpha_{u,i'_2}, \beta_{u,i'_2})] \quad \rangle \; , \end{aligned}$$

przy czym użytkownik $u$ nie zna zastosowanej permutacji, tzn. która trójka odpowiada któremu mapowaniu.

W procedurze kodowania klucza dla użytkowników $\Omega \setminus \Phi$, wykonywane są następujące kroki: $\mathcal{B}$ generuje $x_0, r \in \mathbb{Z}_p$ z rozkładem jednostajnym, oraz wybiera $i \in \{1, 2, 3, 4\}$ z rozkładem jednostajnym. Niech $\Phi_{i,b} = \Omega_{i,b} \cap \Phi$. Nadawca $\mathcal{B}$ koduje klucz $K$ niezależnie dla każdego kubełka $b$. W tym celu:

1. jeśli $|\Phi_{i,b}| < k$, gdzie $k = |\Omega_{i,b}|/2)$, wtedy $\mathcal{B}$ wykonuje procedurę Enc1,

2. jeśli $|\Phi_{i,b}| \geq k$, wtedy $\mathcal{B}$ wykonuje procedurę Enc2.

Nadawca $\mathcal{B}$ wysyła nagłówek $H = \left\langle x_0, g^r, g^{rs}, H_1(i), H_2(i), \ldots, H_{|\Omega|/2k}(i) \right\rangle$ zawierający podnagłówki kubełków $H_b(i)$ skonstruowane procedurami Enc1 lub Enc2.

Procedura Enc1: Jeśli $|\Phi_{i,b}| < |\Omega_{i,b}|/2$, wtedy wykonywane są następujące kroki:

1. $\mathcal{B}$ tworzy zbiór $k$ par $\Psi_b$ w następujący sposób:
   (a) dla każdego wykluczanego użytkownika $u \in \Phi_{i,b}$ (tj. gdy $\Pi_i(u) = b$, $u \in \Phi$) para $(x_{u,i}, g^{rL_{i,\Pi_i(u)}(x_{u,i})})$ jest dodawana do zbioru $\Psi_b$,
   (b) $\mathcal{B}$ dodaje $k - |\Phi_{i,b}|$ par $(x, g^{rL_{i,b}(x)})$ dla $x$ losowanego z rozkładem jednostajnym, i takiego, że dla każdych dwóch różnych par $(x, y), (x', y') \in \Psi_b$ mamy, że $x \neq x'$, oraz $x \notin f_{i,b}(\Omega_{i,b})$. W ten sposób ani $x$ nie jest pseudonimem żadnego uprawnionego użytkownika, ani pseudonimy użytkowników wykluczanych nie pojawiają się więcej niż raz w parze of $\Psi_b$.
2. $\mathcal{B}$ tworzy nagłówek kubełka $H_b(i) = \langle Kg^{rL_{i,b}(x_0)}, \Psi_b \rangle$ .

Procedura Enc2: Jeśli $|\Phi_{i,b}| \geq |\Omega_{i,b}|/2$, wtedy wykonywane są następujące kroki:

1. $\mathcal{B}$ konstruuje zbiór $k+1$ par $XY_b$ w następujący sposób:
   (a) dla każdego użytkownika $u \in \Omega_{i,b} \setminus \Phi_{i,b}$, $\mathcal{B}$ wybiera pseudonim $x_{u,\alpha}$ z rozkładem dyskretnym $\{(x_{u,i}, 1 - p_u), (x_{u,i'}, p_u)\}$, gdzie $\alpha \in \{i, i'\}$ a $x_{u,i}$ jest pseudonimem użytkownika $u$ w mapowaniu $\Pi_i$, podczas gdy $x_{u,i'}$ jest pseudonimem użytkownika $u$ w innym mapowaniu $\Pi_{i'}$, w którym użytkownik $u$ jest przypisany do tego samego kubełka ($\Pi_i(u) = \Pi_{i'}(u) = b$); następnie $\mathcal{B}$ dodaje parę $(x_{u,\alpha}, L_{i,b}(x_{u,\alpha}))$ do zbioru $XY_b$,
   (b) $\mathcal{B}$ wstawia do $XY_b$ pewna liczbę losowych par $(x, y)$ tak aby zbiór $XY_b$ zawierał $k+1$ par i dla różnych par $(x, y), (x', y') \in XY_b$ zachodziło $x \neq x'$.
2. $\mathcal{B}$ oblicza wielomian $W(x)$ interpolacją Lagrange'a zbioru $XY_b$.
3. za pomocą wielomianu $W(x)$, $\mathcal{B}$ konstruuje zbiór $k$ par $\Psi_b$ w następujący sposób:
   (a) dla każdej pary $(x_{u,\alpha}, L_{i,b}(x_{u,\alpha}))$ wybranej w kroku 1a, $\mathcal{B}$ dodaje $(x_{u,\alpha'}, g^{rW(x_{u,\alpha'})})$ do zbioru $\Psi_b$, gdzie $x_{u,\alpha'}$ jest innym pseudonimem użytkownika $u$ przyporządkowanym mu w kubełku $\Pi_\alpha(u)$ (dla mapowania innego niż $\Pi_i$),
   (b) $\mathcal{B}$ wstawia do $\Psi_b$ pary $(x, g^{rW(x)})$ dla $x$ losowanego z rozkładem jednostajnym, tak aby $\Psi_b$ zawierało $k$ par i dla każdych dwóch różnych par $(x, y), (x', y') \in \Psi_b$ zachodziło $x \neq x'$.
4. $\mathcal{B}$ definiuje nagłówek kubełka $H_b(i) = \langle K \cdot g^{rW(x_0)}, \Psi_b \rangle$.

Prawdopodobieństwo $p_u$ w rozkładzie dyskretnym $\{(x_{u,i}, 1-p_u), (x_{u,i'}, p_u)\}$, wykorzystywanym w kroku 1a, jest uznaniowo wybrane przez nadawcę dla każdego użytkownika $u$. Prawdopodobieństwo to określa występowanie pseudonimów użytkownika w nagłówkach i służy zaciemnianiu aktywności użytkownika.

Podczas dekodowania klucza, po otrzymaniu nagłówka $H$, użytkownik $u$ wykonuje następujące kroki algorytmu Decoding, dla wartości $b$ równych $b_1$ oraz $b_2$ określonych w jego kluczu sekretnym $PK_u = \langle [b_1, \ldots], [b_2, \ldots] \rangle$):

1. użytkownik $u$ lokalizuje nagłówek kubełka $H_b = \langle K \cdot g^\eta, \Psi_b \rangle$,
2. dla każdej trójki $(x_{u,i}, \alpha_{u,i}, \beta_{u,i})$ swojego klucza sekretnego $PK_u$ użytkownik $u$ oblicza
$$\omega := (g^r)^{\alpha_{u,i}} \cdot (g^{rs})^{\beta_{u,i}} = (g^r)^{\alpha_{u,i}+s\cdot\beta_{u,i}} = g^{rL(x_{u,i})}),$$
3. $u$ próbuje odkodować klucz $K$ dla każdego $\omega$ uzyskanego w kroku 2:
$$K_b \quad := \quad (K \cdot g^\eta)/LI(x_0, g^r, \Psi_b \cup \{(x_{u,i}, \omega)\}).$$

Zgodnie z powyższą procedurą, jeśli Enc2 została użyta w procesie kodowania klucza, użytkownik uprawniony $u$ uzyskuje
$$LI(x_0, g^r, \Psi_b \cup \{(x_{u,i}, \omega)\} = g^{rL_{i,b}(x_0)} = g^{rW(x_0)}$$
dla swojego pseudonimu $x_{u,i}$, który został wykorzystany do zdefiniowania wielomianu $W(x)$. W tym przypadku $g^{rW(x_0)} = g^\eta$ i użytkownik $u$ wylicza $K_b = K$. Dla błędnych pseudonimów, lub pseudonimów użytkowników wykluczonych, interpolacja Lagrange'a da wynik inny niż $g^{rW(x_0)}$, a dekodowanie klucza nie powiedzie się (tzn. $K_b \neq K$). Jeśli została użyta procedura Enc1, wtedy wykluczony użytkownik nie wykona interpolacji Lagrange'a ponieważ jego udział, znajduje się już w zbiorze $\Psi_b$. Użytkownik uprawniony $u \in \Omega_{i,b}$ może obliczyć $g^{rL_{i,b}(x_0)}$, oraz klucz $K$, ponieważ $g^{rL_{i,b}(x_0)} = g^\eta$.

W omawianej pracy rozważamy adwersarza, który jest zewnętrznym obserwatorem i którego celem jest zdobycie informacji o tożsamościach wykluczanych użytkowników. Adwersarz dostaje zbiór nagłówków, w których użytkownik $u$ zastał wykluczony, a następnie próbuje dopasować do niego właściwy pseudonim. Zauważamy, że taki adwersarz nie rozróżnia, która procedura (Enc1 czy Enc2) została użyta podczas kodowania – nagłówki są nierozróżnialne. Natomiast podczas wykonywania wielokrotnie procedury Enc1 losowy element $r$ randomizuje wykorzystanie tych samych wartości wielomianów. Anonimowość jest zapewniona dzięki dwóm własnościom: 1) każdorazowo podczas użycia procedury Enc2 wykorzystany wielomian jest inny niż $L_{i,b}$; 2) atakujący musi zgadnąć, która z procedur: Enc1 lub Enc2 została wykorzystana. Wynika to z faktu, że nagłówki dla Enc1 oraz Enc2 są obliczeniowo nierozróżnialne przy założeniu trudności problemu DDH. Ta cecha zapewnia, że nie można przeprowadzić ataku na procedurę Enc1 za pomocą techniki z [24]. Wymagają one bowiem $k+1$ właściwych nagłówków. W naszej propozycji prawdopodobieństwo, że wybrany nagłówek został utworzony za pomocą procedury Enc1 wynosi $1/2$, a zatem prawdopodobieństwo, że atakujący wybierze $k+1$ właściwych nagłówków wynosi $(1/2)^{k+1}$.

Dla $k = 100$ jest ono już zaniedbywane. Dla rozważanego modelu adwersarza anonimowość jest rozpatrywana jako niepewność związania wybranego pseudonimu do konkretnego użytkownika w danej sesji. Wybrany pseudonim w różnych sesjach może być wykorzystany dla różnych użytkowników, lub może nie należeć do żadnego wykluczanego użytkownika. W poprzednich schematach, opartych na schemacie dzielenia tajemnic, pseudonim był jednoznacznie przyporządkowany tylko do jednego użytkownika, a w procedurze wykluczania był rozsyłany w nagłówku sesji, co bezpośrednio umożliwiało identyfikację użytkownika.

**Anonimowy sytem BE oparty na złożeniach wielomianów.** Schemat zaproponowany w pracy [A3] rozwiązuje problem zasygnalizowany w pracy [25], dotyczący anonimowego wykluczania $z$ użytkowników ze zbioru maksymalnie $N$, przy zachowaniu niskiej złożoności komunikacyjnej. W pracy tej zaproponowałem pierwszy protokół BE z wykluczaniem zachowujący anonimowość wykluczanych użytkowników, a którego złożoność komunikacyjna jest wprost proporcjonalna do liczby użytkowników wykluczanych (oznaczonej w tej pracy przez $z$). Protokół w ogólnym zarysie opiera się na schemacie [16], który nie jest anonimowy. Aby zapewnić anonimowość w zaproponowanym rozwiązaniu, udziały użytkowników są parametryzowane i zmieniają się dynamicznie w kolejnych sesjach. Jest to nowością w przeciwieństwie do ustalanych jednorazowo wartości w schematach z poprzednich prac. Takie podejście ma ukryć przed obserwatorem możliwość śledzenia i analizowania występowania udziałów wykluczanych w poszczególnych sesjach, co mogłoby się wiązać z analizą wzorców zachowań użytkowników korzystających z usług realizowanych za pomocą schematu. Podobnie jak w pracach [13, 16, 15, 14] schemat zdefiniowany jest w grupie $G$, w której zakładamy trudność DL i opiera się na interpolacji Lagrange'a w wykładniku. Tym razem jednak sekretny wielomian nadawcy $L_t(x) = \sum_i (a_i(t) \cdot x^i)$ ma współczynniki w postaci funkcji zmiennej niezależnej $t$. Ponadto wartości udziałów użytkowników $x_u(t)$ są również zależne od $t$. Dlatego udziały użytkowników wykluczanych, rozgłaszane w nagłówkach, zmieniają się z sesji na sesję. Przyjmujemy, że współczynniki wielomianów wykorzystywanych w tym schemacie są liczbami całkowitymi ze zbioru $\{0, ord(G) - 1\}$.

Podczas rozruchu systemu $B$ tworzy wielomiany współczynników $a_i(t) = \sum_j a_{i,j} t^j$ takie, że and $a_{i,j}$ są wybrane losowo, a następnie tworzy sekretny wielomian $L(t, x) = \sum_i (a_i(t) \cdot x^i)$. Dodatkowo nadawca $B$ tworzy sekretny pomocniczy wielomian $S(t)$ o współczynnikach $\{s\}$ wybranych losowo.

Podczas rejestracji użytkowników $B$ przyporządkowuje niewykorzystany indeks $u$ do nowego użytkownika i wykonuje podstawienie $\Omega \leftarrow \Omega \cup \{u\}$. $B$ tworzy wielomian $x_u(t)$ o współczynnikach $\{x\}$ wybranych losowo. Następnie $B$ oblicza wielomian $L_u(t)$ będący złożeniem $L(t, x_u(t))$ = $\sum_i \left( a_i(t) \cdot \left(x_u(t)\right)^i \right)$ = $\sum_j c_{u,j} t^j$. Dodatkowo nadawca $B$ oblicza wielomiany $P_u(t)$, $Q_u(t)$ takie, że $L_u(t) = P_u(t) + Q_u(t) \cdot S(t)$. Nadawca $B$ przekazuje użytkownikowi $u$ współczynniki: $\{x\}$ wielomianu $x_u(t)$, wpółczynniki $\{p\}$ wielomianu $P_u(t)$ oraz współczynniki w wykładniku $\{g^q\}$ wielomianu $Q_u(t)$. Liczby te są kluczem prywatnym użytkownika $u$. Podczas kodowania klucza nadawca $B$ wybiera losowe $k$, dla którego wylicza klucz sesyjny $K = h(g^k)$. $B$ ustala zbiór użytkowników nieuprawnionych do zdekodowania klucza $\Phi = \{u\}$. Wybiera losowy argument $t_0$, tak aby $x_u(t_0) \neq x_{u'}(t_0)$ dla wszystkich par $u, u'$ użytkowników. Wartość $t_0$ musi być różna w różnych sesjach. $B$ konstruuje zbiór $\Psi$ w następujący sposób:

1. $B$ dla każdego $u$ z $\Phi$ dodaje $x_u(t_0)$ do $\Psi$, tak że $\Psi = \{x_u(t_0) : u \in \Phi\}$
2. $B$ generuje losowy podzbiór $R \subseteq \{0, \ldots ord(G) - 1\} \setminus \Psi$ o mocy $\deg(L(t_0, x)) - |\Phi|$ i aktualizuje zbiór $\Psi$: $\Psi \leftarrow \Psi \cup R$.

Nadawca $B$ wybiera losowo $x_0 \in \{0, \ldots, ord(G)\} \backslash \Psi$, $r \in \{2, \ldots, ord(G)\}$. $B$ oblicza $g^{k+rL(t_0,x_0)}$ i dla każdego $\psi \in \Psi$ oblicza $g^{rL(t_0,\psi)}$. Następnie $B$ generuje nagłówek $H$:

$$H = \left\langle g^{k+rL(t_0,x_0)}, t_0, x_0, g^r, rS(t_0), \left(\psi_1, g^{rL(t_0,\psi_1)}\right), \ldots, \left(\psi_{|\Psi|}, g^{rL(t_0,\psi_{|\Psi|})}\right) \right\rangle,$$

gdzie $\psi_1, \psi_2, \ldots, \psi_{|\Psi|}$ są elementami ze zbioru $\Psi$, a pary $(\psi_i, g^{rL(t_0,\psi_i)})$ w nagłówku są posortowane ze wzgledu na $\psi_i$.

Podczas dekodowania klucza użytkownik $u$ oblicza $x_u(t_0)$, $P_u(t_0)$, $g^{Q_u(t_0)}$ za pomocą współczynników otrzymanych podczas rejestracji. Podstawia $\psi_0 \leftarrow x_u(t_0)$. Jeśli $\psi_0 = \psi_i$ dla niektórych $i \in \{1, \ldots, |\Psi|\}$, to kończy obliczenia nie uzyskawszy pełnego zbioru do przeprowadzenia interpolacji. W przeciwnym wypadku podstawia

$$g^{rL(t_0,\psi_0)} \leftarrow (g^r)^{P_u(t_0)} \cdot \left(g^{Q_u(t_0)}\right)^{rS(t_0)} = g^{rP_u(t_0)+rQ_u(t_0)S(t_0)} = g^{rL(t_0,x_u(t_0))}.$$

Dla wyliczonej pary $\left(\psi_0, g^{rL(t_0,\psi_0)}\right)$ i par z nagłówka $\left(\psi_i, g^{rL(t_0,\psi_i)}\right)$, gdzie $i = 1, \ldots, |\Psi|$, oblicza klucz sesyjny $K = h(g^k)$ dla

$$\begin{aligned} g^k &= \frac{g^{k+rL(t_0,x_0)}}{\prod_{i=0}^{|\Psi|}\left(\left(g^{rL(t_0,\psi_i)}\right)^{\prod_{j=0,j\neq i}^{|\Psi|}\frac{x_0-\psi_j}{\psi_i-\psi_j}}\right)} \\ &= \frac{g^{k+rL(t_0,x_0)}}{g^{r\sum_{i=0}^{|\Psi|}\left(L(t_0,\psi_i)\cdot\prod_{j=0,j\neq i}^{|\Psi|}\frac{x_0-\psi_j}{\psi_i-\psi_j}\right)}} = \frac{g^{k+rL(t_0,x_0)}}{g^{rL(t_0,x_0)}}. \end{aligned}$$

**Anonimowy System BE z wkluczaniem oparty na funkcjach fizycznie nieklonowalnych** Funkcje fizycznie nieklonowalne (ang. Physical Unclonable Functions) - w skrócie PUF - są urządzeniami z interfejsem wejściowym i wyjściowym oraz wewnętrznym fizycznym źródłem losowości, których działanie można opisywać w podobny sposób jak wyrocznię losową ROM. Dla każdej nowej wartości podawanej jako argument, PUF generuje na wyjściu nieprzewidywalną odpowiedź, przy czym dla tych samych wartości wejściowych uzyskujemy takie same odpowiedzi. Źródło losowości urządzeń PUF związane jest z niepowtarzalnymi charakterystykami materiałowymi uzyskiwanymi w sposób niekontrolowany na etapie produkcji. Nieprzewidywalne i niepowtarzalne wariacje fabryczne sprawiają, że urządzenia nie mogą być klonowane nawet przez ich oryginalnego wytwórcę, innymi słowy nie można wytworzyć dwóch różnych takich urządzeń, których odpowiedzi byłyby ze sobą w jakiś sposób skorelowane. Przy projektowaniu systemów wykorzystujących funkcje fizycznie nieklonowalne zakłada się, że z każdym wyprodukowanym urządzeniem PUF związana jest unikalna tablica, w której dla danego nowego wejścia losowane będzie nowe wyjście. W tym kontekście każdy PUF jest unikalnym egzemplarzem jednokierunkowej funkcji modelowanej jako *wyrocznia losowa* ROM.

W pracy [A4] zaproponowałem pierwszy w literaturze przedmiotu system BE, którego bezpieczeństwo oparte jest na funkcjach fizycznie nieklonowalnych PUF. Jak wspomniano powyżej, główne problemy systemów BE opartych na wielomianach, związane są z odpornością na klonowanie urządzeń, atakami koalicji adwersarzy i zapewnieniem prywatności użytkowników. Zaproponowany w [A4] system BE jest odpowiedzią na powyższe problemy: 1) uniemożliwia klonowanie urządzeń, 2) uniemożliwia koalicji adwersarzy dzielenie się swoimi udziałami w celu rekonstrukcji tajnego wielomianu nadawcy 3) zapewnia prywatność uniemożliwiając łączenie rozgłaszanych w kolejnych sesjach udziałów z tożsamościami użytkowników. W szczególności zaproponowałem model bezpieczeństwa uwzględniający wykorzystanie funkcji fizycznie nieklonowalnych. Praca zawiera propozycję architektury urządzeń systemu oraz szczegółowe procedury. Dla twierdzeń o bezpieczeństwie podano ich dowody w zaproponowanym modelu. Przedstawiony system nie wymaga kosztownych obliczeń w wykładniku. Jego wadą są natomiast wymagania związane ze składowaniem sporej ilości danych po stronie nadawcy i użytkownika.

Infrastruktura sprzętowa systemu zakłada, że do dekodera użytkownika $u$ wkładana jest karta, na której znajduje się moduł PUF, oznaczony przez $P_u$, oraz moduł pamięci na dane pomocnicze. Podczas ustalania systemu administrator tworzy losowo $T$ wielomianów stopnia $z$ (gdzie $z$ jest maksymalną liczbą użytkowników wykluczanych w sesji), oraz po dwa parametry $t, t'$ dla każdego z wielomianów. Zbiór $\{L_i(x), t_i, t'_i | i = 1, \ldots, T\}$ stanowi sekret administratora.

Podczas rejestracji użytkownika $u$, dla każdego wielomianu $L_i$ wykonywana jest następująca procedura: Na podstawie wyjścia z PUFa obliczana jest wartość $x_{u,i} = P_u(t_i)$, która jest następnie wykorzystywana do policzenia wartości wielomianu $y_{u,i} = L_i(x_{u,i})$. Następnie wartość ta jest maskowana $y'_{u,i} = y_{u,i} + P_u(t'_i)$ i zapisywana w module pamięci karty użytkownika. W wyniku tej procedury Administrator zapamiętuje wektor $\langle x_{u,1}, \ldots, x_{u,T}\rangle$ dla użytkownika $u$, natomiast w module pamięci karty tego użytkownika zapisany jest wektor $\left\langle y'_{u,1}, \ldots, y'_{u,T}\right\rangle$.

Podczas tworzenia nagłówka rozgłaszania dla sesji o numerze $i$ administrator definiuje zbiór indeksów użytkowników wykluczonych $\Phi \subset \Omega$ i ustala zbiór $X = \{x_{\phi,i} | \phi \in \Phi\}$. Jeśli $|X| < z$ administrator dodaje $z - |X|$ losowych liczb do zbioru $X$, dbając aby wszystkie liczby w $X$ były parami różne. Następnie tworzony jest zbiór $\Psi = \{(x, L_i(x)) | x \in X\}$. Zauważmy, że w zbiorze $\Psi$ brakuje jednego punktu, umożliwiającego interpolację wielomianu $L_i$. Nadawca wybiera losowo $x_i$ takie, że $(x_i \neq x)$ dla każdego $x \in X$, i koduje klucz sesyjny $K_i$ jako $K' = K_i \cdot L_i(x_i)$. Ostatecznie tworzony jest nagłówek $H = \langle i, t_i, t'_i, x_i, \Psi, K'\rangle$, rozgłaszany do wszystkich urządzeń użytkowników. Procedura dekodowania klucza sesyjnego z nagłówka wykonywna w urządzeniu użytkownika $u$ wygląda następująco: Obliczana jest wartość $x_{u,i} = P_u(t_i)$ dla parametru $t_i$. Jeśli $(x_{u,i}, -) \notin \Psi$, wtedy wyliczana jest wartość $y_{u,i} = y'_{u,i} - P_u(t'_i)$. Następnie tworzony jest zbiór interpolacyjny $\Psi' = \Psi \cup \{(x_{u,i}, y_{u,i})\}$, za pomocą którego rekonstruowany jest wielomian $L_i$. Wartość $L_i(x_i)$ dla argumentu $x_i$ z nagłówka, pozwala na odtworzenie klucza sesyjnego $K_i$ równego $K'/L_i(x_i)$.

### 4.3.4 Protokoły identyfikacji

Schemat identyfikacji (ang. Identification Scheme, w skr. IS) umożliwia użytkownikowi udowodnienie swojej tożsamości przed drugą stroną protokołu nazywaną weryfikatorem. Dotychchzas zaproponowano kilka fundamentalnych schematów identyfikacji, np. opartych o kryptosystem RSA (w pracach [26, 27, 28]), lub wykorzystujących problem dyskretnego logarytmu (w pracach [29, 30]). Przykładami specjalizowanych schematów są: schemat z [31] oparty na tożsamościach i bezpieczny w modelu standardowym, lub schemat z [32] wykorzystujący podpisy cyfrowe.

W wielu kryptograficznych schematach IS dowodzący posiada klucz sekretny i udowadnia jego znajomość przed weryfikującym posiadającym odpowiadający klucz publiczny, w taki sposób, że weryfikujący nie uzyskuje żadnej wiedzy na temat klucza sekretnego dowodzącego. Transkrypt trzyrundowego protokołu identyfikacji składa się z trzech komunikatów: zobowiązania, wyzwania, i odpowiedzi. W zobowiązaniu dowodzący wysyła weryfikującemu zobowiązanie do pewnej wartości efemerycznej. W wyzwaniu weryfikator wysyła pewną nieprzewidywalna wartość losową. W odpowiedzi dowodzący wysyła rezultat obliczeń na otrzymanej wartości losowej oraz sekretnym kluczu zamaskowanym wartością efemeryczną do której zobowiązał się w pierwszej wiadomości). Weryfikujący akceptuje dowód, jeśli otrzymana odpowiedź jest "zgodna" z wykonanymi przez niego obliczeniami na zobowiązaniu, wyzwaniu i kluczu publicznym.

**Ataki z ustaloną wartością efemeryczną** Problem z trzyrundowymi schematami identyfikacji pojawia się w sytuacjach, w których wartości efemeryczne wybierane przez dowodzącego mogą zostać ujawnione. Zazwyczaj maskowanie wykorzystywane w komunikacie odpowiedzi jest takie, że bezpieczeństwo sekretu długoterminowego opiera się na tajności klucza efemerycznego, np. odpowiedź jest linową kombinacją wyzwania, klucza długoterminowego i sekretu efemerycznego. Wyciek tego ostatniego trywializuje możliwość obliczenie sekretu długoterminowego. Atak ujawniający klucze efemeryczne może być przeprowadzany za pomocą "błędnych" implementacji schematu dla strony dowodzącej, w szczególności "błędnych" implementacji generatorów liczb pseudolosowych. Poznanie wewnętrznego stanu tych generatorów, lub ich reset za pomocą alternatywnych kanałów, może umożliwić atakującemu poznanie generowanych wartości. W literaturze rozpatrywano podobne scenariusze ataków. Wyciek pojedynczych bitów długoterminowego klucza sekretnego dowodzącego analizowano w [33]. Z kolei problem bezpieczeństwa schematów identyfikacji w scenariuszu resetu stanu generatorów liczb pseudolosowych postawiono w [34]. W pracy [35] przedstawiono kilka ogólnych konstrukcji schematów identyfikacji odpornych na ataki typu *reset*, opartych bądź na istniejących schematach podpisów, bądź na asymetrycznych schematach szyfrowania. Zaproponowane przez mnie schematy z prac [A5, A6, A8, A7] są bezpieczne w modelu, w którym wszystkie wartości efemeryczne wykorzystywane na urządzeniu użytkownika są znane, a nawet ustalane przez adwersarza.

**Zmodyfikowany schemat identyfikacji Schnorra** W pracy [A5] zaproponowałem model bezpieczeństwa dla schematów identyfikacji, w których uwzględniono możliwość dowolnego ustalania kluczy efemerycznych przez adwersarza. Zgodnie z proponowanym modelem schemat jest bezpieczny, jeśli ustalenie kluczy efemerycznych przez adwersarza dla wybranych przebiegów protokołu identyfikacji z udziałem dowodzącego, nie umożliwia adwersarzowi późniejszego przeprowadzenia tego protokołu w roli dowodzącego (personifikacji). W [A5] model ten został wykorzystany do modyfikacji schematu identyfikacji Schnorra [29], którego bezpieczeństwo jest oparte na problemie dyskretnego logarytmu, i który jest jednym z fundamentalnych kryptosystemów stanowiących podstawę bardziej zaawansowanych schematów.

Schemat identyfikacji Schnorra z [29] konstruowany jest w grupie $\mathbb{G} = (p, q, g, G) \leftarrow \mathcal{G}(\lambda)$, w której zakładamy trudność obliczania dyskretnego logarytmu. W procedurze generowania kluczy $\mathsf{KeyGen}()$ tworzony jest klucz prywatny $\mathsf{sk} = a \leftarrow_{\$} \mathbb{Z}_q^*$, oraz klucz publiczny $\mathsf{pk} = A = g^a$. Protokół identyfikacji $\pi(\mathcal{P}(a, A), \mathcal{V}(A))$ pomiędzy dowodzącym $\mathcal{P}(a)$ o tożsamości $\hat{A}$ i weryfikatorem $\mathcal{V}(A)$ ma następujące rundy: $\mathcal{P}$ wybiera sekret efemeryczny $x \leftarrow_{\$} \mathbb{Z}_q^*$, oblicza $X = g^x$ i wysyła $X$ do $\mathcal{V}$. $\mathcal{V}$ wybiera $c \leftarrow_{\$} \mathbb{Z}_q^*$, i wysyła $c$ do $\mathcal{P}$. $\mathcal{P}$ oblicza $s = x + ac$, i wysyła $s$ do $\mathcal{V}$. Weryfikator $\mathcal{V}$ akceptuje tożsamość dowodzącego jeśli $g^s == XA^c$.

Oryginalny schemat Schnorra jest wrażliwy na ataki ujawniające sekret efemeryczny. Z tego powodu w [A5] zaproponowałem schemat zmodyfikowany, w którym ujawnienie, bądź ustalenie klucza efemerycznego przez adwersarza, nie prowadzi do wycieku długoterminowego klucza sekretnego dowodzącego, ani nie umożliwia adwersarzowi personifikacji w późniejszych przebiegach protokołu. W proponowanym modelu bezpieczeństwa, w fazie uczenia, adwersarz w sposób adaptacyjny wybiera sekrety efemeryczne dowodzącego. Niech $\bar{x}$ oznacza sekrety wybrane przez $\tilde{\mathcal{V}}$, a $\mathcal{P}^{\bar{x}}$ oznacza dowodzącego $\mathcal{P}$ wykorzystującego te wartości $\bar{x}$, podczas wykonywania protokołu $\pi(\mathcal{P}^{\bar{x}}(\mathsf{sk}, \mathsf{pk}), \tilde{\mathcal{V}}(\mathsf{pk}, \bar{x}))$. Oznaczmy przez $\mathsf{v}^{\mathcal{P}, \tilde{\mathcal{V}}, \vec{\bar{x}}(\ell)}$ informacje, które $\tilde{\mathcal{V}}$ może uzyskać po przeprowadzeniu $\ell$ wykonań tego protokołu, gdzie $\vec{\bar{x}}(\ell) = \{\bar{x}_1, \ldots, \bar{x}_\ell\}$ są kolejnymi wyborami $\tilde{\mathcal{V}}$.

**Definicja 4.1** (Model bezpieczeństwa *Chosen Prover Ephemeral* – ($\mathsf{CPE}$)). *Niech* $\mathsf{IS} = (\mathsf{ParGen}, \mathsf{KeyGen}, \mathcal{P}, \mathcal{V}, \pi)$ *będzie schematem* $\mathsf{IS}$. *Zdefiniujmy eksperyment bezpieczeństwa* $\mathtt{Exp}_{\mathsf{IS}}^{\mathsf{CPE}, \lambda, \ell}$*:*

Faza $\mathtt{init}$ : *Niech* $\mathsf{params} \leftarrow \mathsf{ParGen}(\lambda)$, $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}()$. *Niech adwersarz* $\mathcal{A}$, *zdefiniowany będzie za pomocą algorytmów* $(\tilde{\mathcal{P}}, \tilde{\mathcal{V}})$ *posiadających klucz publiczny* $\mathsf{pk}$.

Faza $\mathtt{uczenia}$ : $\mathcal{A}$ *uczestniczy w* $\ell$ *of wykonaniach protokołu* $\pi(\mathcal{P}^{\bar{x}_i}(\mathsf{sk}, \mathsf{pk}), \tilde{\mathcal{V}}(\mathsf{pk}, \bar{x}_i))$ *z dowodzącym* $\mathcal{P}^{\bar{x}_i}$, *uzyskując* $\mathsf{v}^{\mathcal{P}, \tilde{\mathcal{V}}, \vec{\bar{x}}(\ell)}$, *gdzie* $\bar{x}_i \in \vec{\bar{x}}(\ell) = \{\bar{x}_1, \ldots, \bar{x}_\ell\}$ *oznacza adaptacyjne wybory* $\tilde{\mathcal{V}}$ *podawane dowodzącemu* $\mathcal{P}^{\bar{x}_i}$ *w* $i$*-tym wykonaniu.*

Faza $\mathtt{ataku}$ "$\mathtt{podszycia}$" : $\mathcal{A}$ *wykonuje protokół* $\pi(\tilde{\mathcal{P}}(\mathsf{pk}, \mathsf{v}^{\mathcal{P}, \tilde{\mathcal{V}}, \vec{\bar{x}}(\ell)}), \mathcal{V}(\mathsf{pk}))$ *z uczciwym weryfikującym.*

*Przewagą adwersarza* $\mathcal{A}$ *w eksperymencie* $\mathtt{Exp}_{\mathsf{IS}}^{\mathsf{CPE}, \lambda, \ell}$ *definiujemy jako prawdopodobieństwo zaakceptowania w fazie personifikacji:*

$$\mathbf{Adv}(\mathcal{A}, \mathtt{Exp}_{\mathsf{IS}}^{\mathsf{CPE}, \lambda, \ell}) = \Pr[\pi(\tilde{\mathcal{P}}(\mathsf{pk}, \mathsf{v}^{\mathcal{P}, \tilde{\mathcal{V}}, \vec{\bar{x}}(\ell)}), \mathcal{V}(\mathsf{pk})) \to 1].$$

*Mówimy, że schemat* $\mathsf{IS}$ *jest bezpieczny jeśli* $\mathbf{Adv}(\mathcal{A}, \mathtt{Exp}_{\mathsf{IS}}^{\mathsf{CPE},\lambda,\ell})$ *jest zaniedbywalną funkcją parametru* $\lambda$.

Zauważmy, że regularny schemat $\mathsf{IS}$ Schnorra nie jest bezpieczny w proponowanym modelu $\mathsf{CPE}$. Znając $\bar{x}$ adwersarz oblicza klucz sekretny $a = (s - \bar{x})/c$ co pozwala na późniejszą personifikację dowodzącego.

Idea modyfikacji jest następująca. Chcemy uniemożliwić obliczenie klucza $a$ dla znanego $c$, $s = x + ac$ oraz ustalonego $x$. Zamiast wysyłać $s$ jawnie, dowodzący wysyła $s$ ukryte w wykładniku $S = \hat{g}^s$, dla nowego generatora $\hat{g} = \mathcal{H}(X|c)$ wyliczonego za pomocą ustalonej funkcji skrótu $\mathcal{H} : \{0,1\}^* \to G$. W konstrukcji schematu wykorzystuje się odwzorowanie dwuliniowe, $\hat{e} : G \times G \to G_T$, za pomocą którego sprawdza się zależność $s = x + ac$ w wykładniku. Zauważmy, że $\hat{e}(S, g)$ = $\hat{e}(\mathcal{H}(X|c)), XA^c)$ ponieważ $\hat{e}(\mathcal{H}(X|c), XA^c)$ = $\hat{e}(\mathcal{H}(X|c)^{x+ac}, g)$. Porównując z oryginalnym schematem, w przedstawionej propozycji mamy jedno potęgowanie i jedno wyliczanie funkcji skrótu więcej po stronie dowodzącego. Po stronie weryfikującego mamy jedno potęgowanie mniej, a dodatkowo wyliczenie funkcji skrótu i porównanie obliczeń dwóch odwzorowań dwuliniowych.

Zmodyfikowany schemat $\mathsf{IS}$ Schnorra jest symulowalny w trybie pasywnym podobnie jak wersja oryginalna. Pasywny obserwator widzi transkrypt $T = (X, c, S)$. W protokole $\pi$ zmienne $x = \log_g X, c$ są niezależne, a razem określają $S = \hat{g}^{x+ac}$ dla ustalonego $a$. Transkrypt ten może być symulowany poprzez wybór $\tilde{s}, \tilde{c}$, a następnie wyliczenie $\tilde{X} = (g^{\tilde{s}}/A^{\tilde{c}})$, oraz $\hat{g} = \mathcal{H}(\tilde{X}|\tilde{c})$ i $\tilde{S} = \hat{g}^{\tilde{s}}$. Dla tak dobranych zmiennych zachodzi: $\hat{e}(\tilde{S}, g)$=$\hat{e}(\mathcal{H}(\tilde{X}|\tilde{c}), \tilde{X}A^{\tilde{c}})$, a krotki $T = (X, c, s)$ oraz $\tilde{T} = (\tilde{X}, \tilde{c}, \tilde{s})$ mają ten sam rozkład.

Analiza bezpieczeństwa proponowanego schematu jest podobna do analizy dotyczącej schematu oryginalnego i wykorzystuje model wyroczni losowej ROM dla funkcji $\mathcal{H}$, oznaczony przez $\mathcal{O}_{\mathcal{H}}$. Zakładamy, że istnieje adwersarz $\tilde{\mathcal{P}}$, którego przewaga nie jest zaniedbywalna. Adwersarz obserwuje wykonanie pewnej liczby przebiegów protokołu, który symulujemy w sposób zaprezentowany powyżej. Następnie w fazie personifikacji wykorzystujemy technikę "nawrotu" (ang. *rewinding technique* ) dla uzyskania dwóch krotek $(X, c_1, S_1, r_1)$, $(X, c_2, S_2, r_2)$, które pozwolą nam złamać problem $\mathsf{GDH}$, czyli obliczyć $g^{\alpha\beta}$ dla zadanych $g^\alpha, g^\beta$. Niech $\mathsf{params} \leftarrow \mathbb{G} = (p, q, g, G)$ będą takie, że zachodzi $\mathsf{GDH}$. Niech $(g^\alpha, g^\beta)$ będą zadanymi liczbami tego problemu. Ustalamy $\mathsf{pk} = g^\alpha$ jako klucz publiczny, podawany adwersarzowi przed przystąpieniem do protokołu. W trakcie protokołu, dla raz ustalonego losowego $x$ takiego, że $X = g^x$ jest wartością wysłaną w pierwszej wiadomości przez $\tilde{\mathcal{P}}$, wykonujemy dwukrotnie kolejne rundy protokołu wybierając za każdym razem, $c_1$ i $c_2$, takie że $X|c_1$ ani $X|c_2$ nie były wcześniej wejściem do wyroczni $\mathcal{O}_{\mathcal{H}}$. Ustalamy $H_1 = \mathcal{O}_{\mathcal{H}}(X|c_1) \leftarrow (g^\beta)^{r_1}$, $H_2 = \mathcal{O}_{\mathcal{H}}(X|c_2) \leftarrow (g^\beta)^{r_2}$ dla $r_1, r_2 \leftarrow \mathbb{Z}_q^*$. Mamy zatem $(X, c_1, S_1, \hat{g}_1, r_1)$ oraz $(X, c_2, S_2, \hat{g}_2, r_2)$ Jeśli $\hat{e}(S_1, g)$ = $\hat{e}(\hat{g}_1, XA^{c_1})$, oraz $\hat{e}(S_2, g)$ = $\hat{e}(\hat{g}_2, XA^{c_2})$, obliczamy: $S_1$ = $(g^{\beta r_1})^x (g^{\beta r_1})^{\alpha c_1}$ oraz $S_2 = (g^{\beta r_2})^x (g^{\beta r_2})^{\alpha c_2}$. W ten sposób $S_1^{{r_1}^{-1}} / {S_2}^{{r_2}^{-1}}$ równa się wartości $(g^\beta)^{\alpha c_1 - \alpha c_2}$, a zatem możemy obliczyć $g^{\alpha\beta}$ wynoszące $(S_1^{{r_1}^{-1}} / {S_2}^{{r_2}^{-1}})^{(c_1 - c_2)^{-1}}$.

Proponowany schemat może być wykorzystany w systemach opartych o dotychczasowy regularny schemat identyfikacji Schnorra, dla których scenariusz wycieku kluczy efemerycznych jest brany pod uwagę. Przykładem może być protokół PACEAA z [1], gdzie oryginalny schemat identyfikacji Schnorra, będący częścią zaprzeczalnego procesu uwierzytelniania, może zostać zastąpiony przez proponowaną zmodyfikowaną wersję. W porównaniu z wcześniejszymi konstrukcjami z [35], propozycja z [A5] zachowująca charakterystyczną konstrukcję oryginalnego schematu identyfikacji Schnorra IS posiada trzy charakterystyczne cechy: 1) jest zdefiniowana w grupach, w których można łatwo definiować protokoły ustalania kluczy sesyjnych oparte na konstrukcji Diffiego-Hellmana; 2) jest trzyrundowa i inicjowana przez zobowiązanie dowodzącego; 3) jest zaprzeczalna dla dowodzącego - tzn. może być symulowana przez weryfikującego bez znajomości długoterminowo klucza sekretnego należącego do dowodzącego.

**Zmodyfikowany schemat identyfikacji Okamoto** W pracy [A6] zaproponowałem modyfikację schematu Okamoto odporną na ustalanie wartości efemerycznych przez adwersarza. Oryginalny schemat Okamoto, podobnie jak schemat Schnorra, konstruowany jest w grupie cyklicznej, w której zakładamy trudność obliczania dyskretnego logarytmu. Niech $g_1 = g$ będzie generatorem rzędu $q$ tej grupy, a $g_2$ jest innym generatorem, takim że $\log_{g_1} g_2$ jest nieznany. W procedurze generowania kluczy $\mathsf{KeyGen}()$ tworzony jest klucz prywatny $\mathsf{sk} = a_1, a_2 \leftarrow_{\$} \mathbb{Z}_q^*$, oraz klucz publiczny $\mathsf{pk} = A$ równy $g_1^{a_1} g_2^{a_2}$. Protokół identyfikacji $\pi(\mathcal{P}(a, A), \mathcal{V}(A))$ pomiędzy dowodzącym $\mathcal{P}(a)$ o tożsamości $\hat{A}$ i weryfikatorem $\mathcal{V}(A)$ ma następujące rundy: $\mathcal{P}$ wybiera sekret efemeryczny $x_1, x_2 \leftarrow_{\$} \mathbb{Z}_q^*$, oblicza $X = g_1^{x_1} g_2^{x_2}$ i wysyła $X$ do $\mathcal{V}$. $\mathcal{V}$ wybiera $c \leftarrow_{\$} \mathbb{Z}_q^*$, i wysyła $c$ do $\mathcal{P}$. Dowodzący $\mathcal{P}$ oblicza $s_1 = x_1 + a_1 c$ oraz $s_2 = x_2 + a_2 c$, a następnie wysyła $s_1, s_2$ do $\mathcal{V}$. Weryfikator $\mathcal{V}$ akceptuje tożsamość dowodzącego jeśli $g_1^{s_1} g_2^{s_2} == X A^c$.

Zauważmy, że podobnie jak schemat IS Schnorra, schemat Okamoto nie jest bezpieczny w proponowanym modelu CPE (Definicja 4.1). Znając wartości efemeryczne $x_1$, $x_2$ adwersarz oblicza klucze sekretne $a_1 = (s_1 - x_1)/c$, $a_2 = (s_2 - x_2)/c$.

Proponowana modyfikacja jest następująca. Zamiast wysyłać $s_1, s_2$ jawnie dowodzący wysyła te wartości ukryte w wykładniku $S_1 = \hat{g}^{s_1}$, $S_2 = \hat{g}^{s_2}$, dla nowego generatora $\hat{g} = \mathcal{H}(X|c)$. Weryfikator sprawdza zależność $s_1 = x_1 + a_1 c$, $s_2 = x_2 + a_2 c$ w wykładniku za pomocą równości: $\hat{e}(S_1, g_1) \cdot \hat{e}(S_2, g_2) = \hat{e}(\mathcal{H}(X|c), X \cdot A^c)$. Zauważmy, że $\hat{e}(S_1, g_1) \cdot \hat{e}(S_2, g_2) = \hat{e}(\mathcal{H}(X|c), g_1^{s_1}) \cdot \hat{e}(\mathcal{H}(X|c), g_2^{s_2}) = \hat{e}(\mathcal{H}(X|c), g_1^{s_1} g_2^{s_2}) = \hat{e}(\mathcal{H}(X|c), g_1^{x_1} g_2^{x_2} g_1^{a_1 c} g_2^{a_2 c}) = \hat{e}(\mathcal{H}(X|c), X A^c)$

Transkrypt protokołu jest symulowalny bez znajomości kluczy $a_1, a_2$ w modelu pasywnym. Losowo z rozkładem jednostajnym wybierane są: $s_1, s_2, c$. Oblicza się $X = (({g_1}^{s_1} {g_2}^{s_2})/A^c)$, a następnie $\hat{g} = \mathcal{H}(X|c)$ oraz $S_1 = \hat{g}^{s_1}$, $S_2 = \hat{g}^{s_2}$. Dla tak wygenerowanego transkryptu $(X, S_1, S_2, c)$ weryfikacja daje wynik pozytywny: $\hat{e}(S_1, g_1) \hat{e}(S_2, g_2) = \hat{e}(\hat{g}, X A^c)$.

Zmodyfikowany schemat Okamoto IS jest symulowalny w modelu CPE z wykorzystaniem programowalnej wyroczni losowej $\mathcal{O}_{\mathcal{H}}$. Algorytm symulatora $\mathcal{S}_{\mathsf{IS}}^{\mathsf{CPE}, \pi}()$ jest zdefiniowany w następujący sposób: Dla danych $(g_1, g_2, A)$ wybiera się $a_2 \leftarrow_{\$} \mathbb{Z}_q^*$, oraz oblicza ${g_1}^{a_1} = A/({g_2}^{a_2})$.

Ustala się tablicę wyroczni losowej $\mathcal{O}_{\mathcal{H}}()$ z trzema kolumnami $I, H, r$: odpowiednio dla wejścia, wyjścia i maski wykładnika. W każdym zapytaniu $\mathcal{O}_{\mathcal{H}}(I_i)$ sprawdza się czy wejście $I_i$ zapisane jest już w tablicy - jeśli tak, to zwracane jest odpowiadające mu wyjście $H_i$. W przeciwnym wypadku losuje sie maskę $r_i \leftarrow_{\$} \mathbb{Z}_q^*$, oblicza $H_i = g^{r_i}$, wstawia się krotkę $(I_i, H_i, r_i)$ do tablicy, oraz zwraca wartość $H_i$. Dla ustalonych przez adwersarza wartości $(\bar{x}_1, \bar{x}_2)$ wylicza się zobowiązanie $\tilde{X} = {g_1}^{\bar{x}_1} {g_2}^{\bar{x}_2}$. Po otrzymaniu wyzwania $\tilde{c}$ wywołuje się wyrocznię $\mathcal{O}_{\mathcal{H}}(\bar{X}|\tilde{c})$. Sprawdza się tablicę wyroczni $\mathcal{O}_{\mathcal{H}}$ dla wejścia $\bar{X}|\tilde{c}$, lokalizuje odpowiadające mu wartości $g^r$ oraz $r$, i zwraca jako odpowiedź wyroczni wartość $\hat{g} = g^r$. Mając $r$ oblicza się $S_1 = {g_1}^{r\bar{x}_1}(A/{g_2}^{a_2})^{rc}$ równe wartości ${g_1}^{r\bar{x}_1}({g_1}^{a_1})^{rc} = \hat{g}^{\bar{x}_1+a_1c}$ oraz $S_2 = {g_2}^{r\bar{x}_2+a_2rc} = \hat{g}^{\bar{x}_2+a_2c}$. Zauważmy, że wtedy weryfikacja daje wynik pozytywny: $\hat{e}(\tilde{S}_1, g_1)\hat{e}(\tilde{S}_2, g_2) = \hat{e}(\hat{g}, XA^{\tilde{c}})$.

Bezpieczeństwa proponowanej modyfikacji dowodzi się poprzez redukcjędo problemu SDH. Zakładając istnienie efektywnego algorytmu adwersarza, wykorzystuje się go jako podprocedurę do złamania problemu GDH zadanego wartościami $g^\alpha, g^\beta$, czyli do wyliczenia wartości $g^{\alpha\beta}$. Podczas konstrukcji systemu wybiera się $(a_2, w) \leftarrow_{\$} \mathbb{Z}_q^*$. Ustala $g_1 = g, g_2 = g^w, \mathsf{pk} = g^\alpha = A$. Wtedy mamy ${g_1}^{a_1} = A/({g_2}^{a_2}) = A/(g^{a_2w})$. Adwersarz $\mathcal{A}$, dostaje $\mathsf{pk}={g_1}^{a_1}{g_2}^{a_2}=g^\alpha=A$. W modelu z wyrocznią, w fazie uczenia, pozwala się adwersarzowi pełnić rolę weryfikatora, dla przebiegów protokołu wykonywanego symulatorem $\mathcal{S}_{\mathsf{IS}}^{\mathsf{CPE},\pi}()$. Następnie w fazie ataku, w dwóch przebiegach protokołu, uzyskujemy dwie krotki transkryptu $(X, c, S_1, S_2)$, $(X, c', S_1', S_2')$, dla takiego samego zobowiązania $X$. Wyzwania $c$ oraz $c'$, wybierane są tak aby wartości $X|c$ oraz $X|c'$ były różne od tych rejestrowanych na wejściu wyroczni $\mathcal{O}_{\mathcal{H}}$ w fazie uczenia. Wtedy na wyjściu $\mathcal{O}_{\mathcal{H}}(X|c)$ ustala się wartość $\hat{g} = (g^\beta)^r$, a na wyjściu $\mathcal{O}_{\mathcal{H}}(X|c')$ wartość $\hat{g}'= (g^\beta)^{r'}$ dla $r, r' \leftarrow_{\$} \mathbb{Z}_q^*$. Jeśli adwersarz zostaje zaakceptowany w tych dwóch przebiegach protokołu, tzn. jeśli: $\hat{e}(S_1, g_1)\hat{e}(S_2, g_2)$ jest równe $\hat{e}(\hat{g}, XA^c)$ oraz $\hat{e}(S_1', g_1)\hat{e}(S_2', g_2) = \hat{e}(\hat{g}, XA^{c'})$ to wnioskujemy następująco. Niech $(x_1, x_2)$ and $(a_1, a_2)$ oznaczają wartości efemeryczne i klucze prywatne wykorzystywane przez adwersarza w pierwszym przebiegu $X = {g_1}^{x_1}{g_2}^{x_2} = g^{x_1+wx_2}$, $A = {g_1}^{a_1}{g_2}^{a_2} = g^{a_1+wa_2}$, dla których wyliczono $S_1, S_2$. Podobnie niech $(x_1', x_2')$ oraz $(a_1', a_2')$ są wartościami wykorzystywanymi przez adwersarza w drugim przebiegu: $X = {g_1}^{x_1'}{g_2}^{x_2'} = g^{x_1'+wx_2'}$, $A = {g_1}^{a_1'}{g_2}^{a_2'} = g^{a_1'+wa_2'}$, dla których wyliczono $S_1', S_2'$. Mamy wtedy: ${S_1}^{(r^{-1})}/{S_1'}^{(r'^{-1})} = (g^\beta)^{(x_1+a_1c)-(x_1'+a_1'c')}$ oraz ${S_2}^{(r^{-1})}/{S_2'}^{(r'^{-1})} = (g^\beta)^{(x_2+a_2c)-(x_2'+a_2'c')}$. Zatem $({S_1}^{(r^{-1})}/{S_1'}^{(r'^{-1})})({S_2}^{(r^{-1})}/{S_2'}^{(r'^{-1})})^w$ jest równe $(g^\beta)^{(x_1+a_1c)-(x_1'+a_1'c')}(g^\beta)^{w(x_2+a_2c)-w(x_2'+a_2'c')}$, co jest równe $(g^\beta)^{(x_1+wx_2)-(x_1'+wx_2')}(g^\beta)^{c(a_1+wa_2)-c'(a_1'+wa_2')}$, co wynosi $(g^\beta)^{(c-c')\alpha}$. Zatem obliczyliśmy $g^{\beta\alpha}$ jako $(({S_1}^{(r^{-1})}/{S_1'}^{(r'^{-1})})({S_2}^{(r^{-1})}/{S_2'}^{(r'^{-1})})^w)^{((c-c')^{-1})}$.

**Protokół anonimowej weryfikacji poświadczeń atrybutów użytkownika** W pracy [A7] zaproponowałem schemat anonimowego weryfikowania poświadczeń atrybutów bezpieczny w modelu z wyciekiem wartości efemerycznych z urządzenia użytkownika. Schemat weryfikowania anonimowych poświadczeń umożliwia użytkownikowi udowodnienie przed weryfikującym posiadania poświadczeń dla posiadanych atrybutów bez ujawniania swojej tożsamości. W schemacie tym użytkownik dysponuje pewną grupą atrybutów, dla których uzyskuje od urzędu poświadczającego certyfikat ich posiadania. W ramach protokołu weryfikacji użytkownik dowodzi, że posiada wydane certyfikaty w taki sposób, że weryfikujący uznaje dowód, ale nie jest w stanie zidentyfikować użytkownika. Dodatkową cechą, spełnianą w wybranych schematch, jest trudność powiązania tożsamości z wcześniejszymi wykonaniami tego protokołu. W omawianej pracy [A7] zaproponowano modyfikację schematu [36], którego protokoły zawierają konstrukcje zbliżone ze schematów [29, 30], a zatem podatnych na ataki ujawniające wartości efemeryczne. W schemacie wyróżnione są trzy strony: użytkownik posiadający atrybuty $\{m_i\}_1^l$, wydawca posiadający zestawy kluczy prywatnych i publicznych $(x, y, \{z_i\}_1^l)$ i $(X = g^x, Y = g^y, \{Z_i = g_i^z\}_1^l)$ odpowiednio, oraz weryfikujący znający klucze publiczne wydawcy.

Protokół uzyskiwania certyfikatu, wykonywany pomiędzy użytkownikiem a wystawcą, składa się z dwóch części. W pierwszej części użytkownik wykonuje protokół z wiedzą zerową znajomości swoich atrybutów: Przesyła do wystawcy wartości $M = g^{m_0}\Pi_{i=1}^{l} Z_i^{m_i}$ oraz $T = g^{r_0}\ \Pi_{i=1}^{l} Z_i^{r_i}$ wyliczone dla wartości efemerycznych $(r_0, \dots r_l) \leftarrow_{\$} \mathbb{Z}_q^*$. Wystawca odsyła losowe wyzwanie $c \leftarrow_{\$} \mathbb{Z}_q^*$, na które użytkownik odpowiada wartościami $s_i = r_i - cm_i$ dla każdego $i \in \{0, \dots, l\}$. Wystawca akceptuje dowód jeśli $T \overset{?}{=} M^c\ g^{s_0}\ \Pi_{i=1}^{l} Z_i^{s_i}$. Zwróćmy uwagę, że ta część jest rozszerzoną wersją protokołu identyfikacji Okamoto, w której tajne atrybuty $m_i$ maskowane są wartościami $r_i$ w równaniach liniowych. Rozwiązanie to jest zatem podatne na ataki ze strony adwersarza znającego wartości efemeryczne. W końcowej części wystawca wydaje użytkownikowi certyfikat będący podpisem CL [36] pod wiadomością $M$: Wylicza $a_0 \leftarrow_{\$} \mathbb{Z}_q^*$, $A_0 = g^{a_0}$, $\forall_{i \in \{1,\dots,l\}}\ A_i = A_0^{z_i}$, $\forall_{i \in \{0,\dots,l\}}\ B_i = A_i^y$, $C = A_0^x\ M^{a_0 xy}$. Przesyła użytkownikowi wartości $\{A_i\}_0^l$, $\{B_i\}_0^l$, $C$ stanowiące certyfikat.

W protokole weryfikowania atrybutów użytkownik losuje $(r', r'', r_a, r_0, \dots, r_l) \leftarrow_{\$} \mathbb{Z}_q^*$, którymi maskuje części certyfikatu atrybutów: $\tilde{A}_i = A_i^{r'}, \tilde{B}_i = B_i^{r'}, \tilde{C} = C^{r'r''}$ dla każdego $i \in \{0, \dots, l\}$; oblicza zobowiązanie $\hat{t} = \widehat{e}(X, \tilde{A}_0)^{r_a}\ \Pi_{i=0}^{l} \widehat{e}(X, \tilde{B}_i)^{r_i}$ i wysyła $\{\tilde{A}_i\}_0^l, \{\tilde{B}_i\}_0^l, \tilde{C}, \hat{t}$ do weryfikatora. Weryfikator po wstępnym sprawdzeniu poprawności maskowania $e(\tilde{A}_0, Z_i) \overset{?}{=} \widehat{e}(g, \tilde{A}_i)$ oraz $\widehat{e}(\tilde{A}_i, Y) \overset{?}{=} \widehat{e}(g, \tilde{B}_i)$ wysyła wyzwanie $c \leftarrow_{\$} \mathbb{Z}_q^*$ do użytkownika. Użytkownik wylicza $s_a = r_a - cr''$ oraz $s_i = r_i - cm_i r''$ dla i $\in \{0, \dots, l\}$, i wysyła $s_a, \{s_i\}_0^l$ do weryfikatora, który sprawdza zgodność tych wartości ze zobowiązaniem: $\hat{t} \overset{?}{=} \widehat{e}(g, \tilde{C})^c \widehat{e}(X, \tilde{A}_0)^{s_a} \Pi_{i=0}^{l} \widehat{e}(X, \tilde{B}_i)^{s_i}$. Podobnie jak przy wystawianiu certyfikatu, także i tutaj wartości $m_i$ w równaniach liniowych maskowane są wartościami efemerycznymi $r'', r_a, r_i$, których wyciek ujawnia atrybuty użytkownika.

Proponowana w pracy [A7] modyfikacja protokołów wystawiania certyfikatu oraz weryfikacji atrybutów polega na wyliczaniu i przesyłaniu równań liniowych $s_a, s_i$ w wykładniku dla nowych generatorów podawanych użytkownikowi w wyzwaniu. Przy wystawianiu certyfikatu wystawca w wyzwaniu wysyła wartości $c, \tilde{g} = g^{\omega}$ dla $(c, \omega) \leftarrow_{\$} \mathbb{Z}_q^*$. W odpowiedzi użytkownik wysyła $S_i = \tilde{g}^{r_i - cm_i}$, a wystawca sprawdza równość $\widehat{e}(\tilde{g}, T/M^c) \overset{?}{=} \widehat{e}(S_0, g)\ \prod_{i=1}^{l} \widehat{e}(S_i, Z_i)$. Z kolei przy weryfikacji certyfikowanych atrybutów weryfikujący w wyzwaniu wysyła wartości $c, \overline{X} = X^{\omega}$ dla $(c, \omega) \leftarrow_{\$} \mathbb{Z}_q^*$. W odpowiedzi użytkownik wysyła $s_a = r_a - cr''$, $S_i = \overline{X}^{\,r_i - cm_i r''}$, a weryfikujący sprawdza równość $\hat{t}^{\omega} \overset{?}{=} \widehat{e}(g^{\omega c}, \tilde{C}) \widehat{e}(\overline{X}, \tilde{A}_0)^{s_a} \Pi_{i=0}^{l} \widehat{e}(S_i, \tilde{B}_i)$.

Bezpieczeństwo protokołu wystawiania certyfikatów opiera się na problemie GDH. Rozumowanie dowodowe jest zbliżone tego przedstawionego w pracach [A5, A6], tzn. algorytm adwersarza wykonujący z sukcesem protokół uzyskiwania certyfikatu bez znajomości wymaganych atrybutów $m_i$ może zostać wykorzystany do złamania wybranej instancji problemu GDH. Z kolei bezpieczeństwo protokołu weryfikacji atrybutów opiera się na zmodyfikowanym założeniu LRSW [37], a algorytm adwersarza wykonujący z sukcesem protokół strony użytkownika nie posiadającego certyfikowanych atrybutów może zostać wykorzystany do złamania tego założenia. Proponowane modyfikacje zachowują pożądane właściwości protokołów oryginalnych: protokoły są zaprzeczalne dla użytkownika, a protokół weryfikacji nie pozwala związać wielu wykonań z konkretną tożsamością.

**Silnie zaprzeczalne rozszerzenia dla protokołów trzyrundowych** W pracy [A8] zaproponowałem rozszerzenia dla trzyrundowych protokołów identyfikacji, uniemożliwiające adwersarzowi, pełniącemu rolę weryfikatora, ominięcie własności zaprzeczalności tych protokołów. Protokoły tego typu, np.: [29, 30], lub [A5, A6], są zaprzeczalne jedynie w przypadku tzw. uczciwego weryfikatora, który wyzwanie generuje w sposób zgodny ze schematem, czyli losowy. W tym przypadku zaprzeczalność związana jest z istnieniem algorytmów symulujących, za pomocą których, bez klucza tajnego na wejściu, można wygenerować transkrypty tych protokołów, o rozkładzie identycznym jak transkrypty oryginalnie. Nieuczciwy weryfikujący może jednak odstąpić od procedury schematu i generować wyzwania będące wynikiem transformacji Fiata-Shamira na zobowiązaniach użytkowników dowodzących swej tożsamości. W ten sposób transkrypt zaprzeczalnego protokołu identyfikacji przekształca się w niezaprzeczalny podpis użytkownika - będący dowodem jego interakcji z weryfikatorem. W pracy zaproponowałem silniejszy (w stosunku do modelu z [A5, A6]) model bezpieczeństwa, w którym adwersarz bez klucza tajnego nie powinien zostać uwierzytelniony, nawet wtedy, gdy wcześniej obserwuje wykonanie protokołu za pomocą urządzenia, z którego wypływają wartości efemeryczne, a dodatkowo, w fazie ataku poznaje wartości efemeryczne weryfikującego natychmiast po ich wygenerowaniu.

Intuicyjnym podejściem do tego problemu jest utworzenie własnego zobowiązania przez weryfikatora w dodatkowym, pierwszym kroku protokołu. Losowość tego zobowiązania powoduje, że ukryta wartość efemeryczna pozostaje nieznana do momentu jej odkrycia, co następuje dopiero po otrzymaniu przez weryfikatora zobowiązania użytkownika. omawianej pracy zaproponowane są dwa specyficzne protokoły realizujące to podejście.

W podejściu opartym na deterministycznym schemacie szyfrowania zakładamy, że weryfikator posiada długoterminowy klucz tajny $\mathsf{se}$, przechowywany w bezpiecznym module pamięci, a odpowiadający klucz publiczny $\mathsf{pe}$ jest znany dowodzącemu. Niech $\mathsf{IS} = (\mathsf{ParGen}, \mathsf{KeyGen}_{\mathcal{P}}, \mathcal{P}, \mathcal{V}, \pi)$ będzie schematem identyfikacji, w którym $(X, c, S)$ oznacza zobowiązanie, wyzwanie i odpowiedź. Niech $\mathcal{H}$ oznacza bezpieczną funkcje skrótu w przestrzeń wyzwań. Niech $(\mathsf{KG}_{\mathsf{E}}, \mathcal{E}, \mathcal{D})$ oznacza wybrany deterministyczny system szyfrowania. Zmodyfikowany protokół identyfikacji $\pi(\mathcal{P}(\mathsf{sk}, \mathsf{pe}), \mathcal{V}(\mathsf{pk}, \mathsf{se}))$ wygląda następująco. Weryfikator $\mathcal{V}$ zobowiązuje się do nieznanej wartości jawnej w losowym szyfrogramie $\hat{c} \leftarrow_{\$} C$. Dowodzący $\mathcal{P}$ przygotowuje własne zobowiązanie $X$. Dopiero po otrzymaniu $X$ weryfikujący $\mathcal{V}$ deszyfruje $m = \mathcal{D}(\mathsf{se}, \hat{c})$ i wysyła $m$ do dowodzącego wraz z losowym bitem $b \leftarrow_{\$} \{0, 1\}$. Dowodzący sprawdza czy $\mathcal{E}(\mathsf{pe}, m) \overset{?}{=} \hat{c}$ upewniając się, że $m$ nie jest zależne od $X$, oblicza $c = \mathcal{H}(m, b)$, a następnie przygotowuje $S$ zgodnie z protokołem oryginalnym. W ten sposób weryfikujący ma pewność, że zobowiązanie $X$ nie zostało wyliczone jako funkcja wartości $m$, a dowodzący upewnia się, że $m$ nie jest zależne od $X$.

W podejściu opartym na *dowodzie obliczalności* zakłada się, że urządzenie weryfikatora jest wydajniejsze niż urządzenie dowodzącego, a zatem szybciej może rozwiązać wygenerowany pseudolosowo ciąg problemów $\mathsf{P}$ o zadanej złożoności obliczeniowej. Zakładamy przy tym, że rozwiązanie instancji problemu $X_i$ wygenerowanego funkcją $\mathcal{G}(\mathsf{P}, w_i)$ z zarodka $w_i$, trwa dużo dłużej niż weryfikacja poprawności tego rozwiązania $\mathsf{Ver}(\mathsf{P}, X_i, \varsigma_i)$. Zmodyfikowany protokół identyfikacji $\pi(\mathcal{P}(\mathsf{sk}), \mathcal{V}(\mathsf{pk}))$ wygląda następująco. Weryfikujący $\mathcal{V}$ losuje i wysyła do dowodzącego zarodek $w$, wykorzystywany do deterministycznego generowania pewnej klasy problemów obliczeniowych $\mathsf{Gen}(\mathsf{P}, w)$ w taki sposób, że rozwiązanie $\varsigma_i$ $i$-tej instancji problemu jest wykorzystane jako zarodek do generowania problemu następnego. Po odebraniu zobowiązania od dowodzącego, weryfikujący $\mathcal{V}$ kończy generowanie problemów uzyskując ciąg rozwiązań $\langle\varsigma_i\rangle_i^n$, który wysyła do dowodzącego. Po pozytywnym zweryfikowaniu ich poprawności, $\mathcal{P}$ wylicza wyzwanie $c = \mathcal{H}(\langle\varsigma_i\rangle_i^n)$, a następnie przygotowuje $S$ zgodnie z protokołem oryginalnym. Zakładając większą wydajność weryfikatora dowodzący nie może wyliczyć swojego zobowiązania przed otrzymaniem wyzwania. Z drugiej strony ma pewność, że $\langle\varsigma_i\rangle_i^n$ jest ciągiem zależnym od $w$, a zatem wyzwanie $c$ nie jest zależne od $X$.

**Protokoły identyfikacji sektorowej** W pracy [A9] zaproponowano protokół identyfikacji użytkownika w wielu niezależnych realizacjach schematu $\mathsf{IS}$ (tzw. *sektorach*), za pomocą operacji wykonywanych przy pomocy jednego klucza prywatnego. Protokół ten jest modyfikacją niemieckiego protokołu [3], który może być implementowany w elektronicznych dowodach tożsamości do identyfikacji obywatela w wielu niezależnych podsystemach, np. urzędach skarbowych, placówkach służby zdrowia, itp. Istotną cecha tego protokołu jest zapewnienie prywatności użytkownika w taki sposób, aby jego identyfikacja w jednym sektorze nie pozwalała na dodatkową identyfikację w innym sektorze, pomimo wykorzystania tego samego klucza prywatnego. Konstrukcja tego protokołu umożliwia jego implementację na urządzeniach o ograniczonej mocy obliczeniowej. Zaproponowana modyfikacja uwzględnia sytuacje, w której prawa do identyfikacji w danym sektorze mogą zostać cofnięte. Realizowane jest to za pomocą tzw. białych list - dla użytkowników uprawnionych, niezależnych dla każdego sektora. Protokół jest odporny na ataki adwersarza próbującego związać działania tego samego użytkownika w różnych sektorach, w sytuacji gdy białe listy z tych sektorów, klucze publiczne oraz transkrypcje wykonywanych protokołów są dla adwersarza widoczne.

Wyróżniono trzy rodzaje uczestników protokołu: 1) *urząd identyfikacyjny* przechowuje klucze $r_j$ dla każdego sektora $S_j$; 2) *urząd sektora* $S_j$ posiada parę kluczy (prywatny/publiczny) asymetrycznego kryptosystemu $E$, który jest wykorzystywany przy szyfrowaniu wiadomości wysłanych do $S_j$, gdzie $K_j$ jest kluczem publicznym. Dodatkowo $S_j$ posiada tajny klucz $R_j$ oraz publiczny klucz $Y_j = g^{r_j R_j}$; 3) Użytkownik $U_i$ posiada tajny klucz $x_i$, oraz główny klucz publiczny $y_i = g^{x_i}$. W systemie utrzymywane są następujące listy: a) lista użytkowników $(U_i, y_i)$ utrzymywana przez *urząd identyfikacyjny*; b) lista sektorów w formacie $(S_j, Y_j, K_j)$; c) lista użytkowników w sektorach: w postaci tzw. *białych list* $W_j = (y_{i,j})$ kluczy publicznych użytkowników uprawnionych w sektorach, gdzie $y_{i,j} = Y_j^{x_i}$ jest kluczem publicznym $i$-tego użytkownika w sektorze $S_j$.

Podczas tworzenia list sektorów *urząd identyfikacyjny* wybiera losowe $r_j$, wylicza $z_j = g^{r_j}$ i wysyła $z_j$ do $S_j$. Sektor $S_j$ wybiera losowe $R_j$, wylicza $Y_j = z_j^{R_j}$ i wysyła $Y_j$ do *urzędu identyfikacyjnego*. Urząd identyfikacyjny wstawia krotkę $(S_j, Y_j, K_j)$ do listy sektorów i wystawia certyfikat wiążący $S_j$ z $Y_j$. Przy tworzeniu kluczy w sektorze $S_j$ dla użytkownika $U_i$ posiadającego klucze główne $(x_i, y_i = g^{x_i})$ *urząd identyfikacyjny* wylicza $y_i^{r_j}$ i przesyła tą wartość do $S_j$, który wylicza $y_{i,j} = (y_i^{r_j})^{R_j}$.

Protokół uwierzytelnionej wymiany klucza w sektorze wygląda następująco: Użytkownik $U_i$ otrzymuje certyfikat sektora $S_j$ zawierający klucz $Y_j$. $U_i$ oblicza swój klucz publiczny $y_{i,j} = Y_j^{x_i}$ w sektorze $S_j$. $U_i$ wybiera losowe $v$ i szyfruje je za pomocą klucza $K_j$ otrzymując $e = E_{K_j}(v)$. $U_i$ wysyła do $S_j$ szyfrogramy $e$ oraz $E_{K_j}(y_{i,j}^v)$ wraz z żądaniem dostępu. $S_j$ po odszyfrowaniu szyfrogramów sprawdza czy $y_{i,j} == (y_{i,j}^v)^{v^{-1}}$. Jeśli nie, dostęp nie jest przyznawany. W przeciwnym przypadku $S_j$ generuje losowo $u_1, u_2$, wylicza $h_1 = Y_j^{u_1}$, $h_2 = Y_j^{u_2}$ i wysyła do $U_i$ wartości $h_1, h_2$ wraz z nie-interaktywnym dowodem z wiedzą zerową znajomości logarytmu $\log_{Y_j} h_2$. Następnie oblicza klucz sesyjny $K = (y_{i,j}^{u_1})^v$ i jednorazowy token $S = (y_{i,j}^{u_2})^v$. $U_i$ po pozytywnym zweryfikowaniu dowodu dotyczącego $\log_{Y_j} h_2$ oblicza klucz sesyjny $K = (h_1^{x_i})^v$ i jednorazowy token $S = (h_2^{x_i})^v$. Dalsza komunikacja jest szyfrowana kluczem $K$. W swoim pierwszym komunikacie $U_i$ wysyła token $S$, który jest porównywany z wartością wyliczoną lokalnie przez $S_j$. Z kolei $S_j$ w swoim pierwszym komunikacie wysyła $u_2$, a $U_j$ sprawdza równość $h_2 = Y_j^{u_2}$. Zauważmy, że klucz sesyjny $K$ wyliczany niezależnie dla każdej ze stron ma taka samą wartość $K = h_1^{x_i}$ równe $(Y_j^{u_1})^{x_i} = (Y_j^{x_i})^{u_1} = y_{i,j}^{u_1}$. Podobnie jest dla tokena $S$. Jeśli zachodzi konieczność wykluczenia użytkownika posiadającego sektorowy klucz publiczny $y'_{i,j}$ wtedy urząd $S_j$ wylicza $(y'_{i,j})^{(R_j^{-1})}$ i wysyła tą wartość do *urzędu identyfikacyjnego*, który z kolei wylicza $y'_i = ((y'_{i,j})^{(R_j^{-1})})^{(r^{-1})}$ i identyfikuje posiadacza tego klucza publicznego.

Zaproponowany protokół jest odporny na następujące ataki: 1) Atak polegający na uzyskaniu tajnego klucza $x_i$ użytkownika $U_i$ przez adwersarza posiadającego wszystkie pozostałe klucze prywatne, wszystkie dane publiczne oraz transkrypty wykonywanych protokołów. 2) Atak polegający na uzyskaniu tajnego klucza $R_j$ sektora $S_j$ przez adwersarza posiadającego wszystkie pozostałe klucze prywatne, wszystkie dane publiczne oraz transkrypty wykonywanych protokołów. 3) Atak polegający na uwierzytelnieniu się w imieniu $U_i$ w sektorze $S_j$ przez adwersarza posiadającego wszystkie pozostałe klucze prywatne poza $x_i, R_j$, wszystkie dane publiczne oraz transkrypty wykonywanych protokołów. 4) Atak polegający na uwierzytelnieniu się w imieniu $S_j$ przed użytkownikiem $U_i$ przez adwersarza posiadającego wszystkie pozostałe klucze prywatne poza $x_i, R_j$, wszystkie dane publiczne oraz transkrypty wykonywanych protokołów. 5) Atak na prywatność użytkowników w sektorach polegający rozstrzygnięciu czy dwa wybrane z dwóch sektorów $S_j, S_{j'}$ klucze publiczne $y_{i,j}, y_{i',j'}$ należą do tego samego użytkownika, czyli czy $i = i'$. Atak ten jest uściślony w następujący sposób. Zakłada się, że jest tylko dwóch użytkowników $U_1, U_2$ we wszystkich sektorach. Adwersarz nie posiada jedynie prywatnych kluczy $x_1, x_2$, oraz tajnych kluczy $r_j$ będących w posiadaniu urzędu identyfikacyjnego. Celem adwersarza jest wskazanie publicznego klucza użytkownika $U_1$ w wybranym sektorze $S_1$. Dowód odporności na ten atak opiera się na trudności Wiązanego Problemu Diffiego-Hellmana postawionego w pracy (ang. Linking Diffie-Hellmana Problem - LDHP). Dla tego problemu w pracy wykazano, że jeśli: $\mathbb{G} \leftarrow_{\$} \mathcal{G}(\lambda)$, $a \leftarrow_{\$} \mathbb{Z}_q^*, b \leftarrow_{\$} \mathbb{Z}_q^*, r \leftarrow_{\$} \mathbb{Z}_q^*, D_0 = (\mathbb{G}, g^a, g^b, g^r, g^{ra}, g^{rb}), D_1 = (\mathbb{G}, g^a, g^b, g^r, g^{rb}, g^{ra})$, to dla każdego probabilistycznego algorytmu działającego w czasie wielomianowym $\mathcal{A}_{\mathsf{LDHP}}$ zachodzi: $|\Pr[\mathcal{A}_{\mathsf{LDHP}}(D_0) = 0] - \Pr[\mathcal{A}(D_1) = 0]| \leq \epsilon_{\mathsf{LDHP}}(\lambda)$, gdzie $\epsilon_{\mathsf{LDHP}}(\lambda)$ jest zaniedbywalne.

### 4.3.5 Protokoły uwierzytelnionego ustalania klucza sesyjnego

W protokołach uwierzytelniania i wymiany klucza (ang. Authenticated Key Establishment - AKE) wyróżnia się dwie strony, których celem jest wzajemna identyfikacja i ustalenie wspólnego klucza symetrycznego, za pomocą którego szyfrowana będzie komunikacja. Zazwyczaj strony protokołu posługują się certyfikowanymi kluczami publicznymi, które wykorzystywane są we wzajemnej identyfikacji. Przyjmuje się, że protokoły AKE wykonywane są w niezabezpieczonym kanale transmisyjnym, a ustalony klucz pozwala kanał ten zabezpieczyć. Modele bezpieczeństwa dla protokołów AKE uwzględniają obecność adwersarza, który obserwując, bądź ingerując w przekazywane wiadomości próbuje: uzyskać uzgadniany klucz sesyjny (atak na tajność klucza), podszyć się pod którąś ze stron (atak personifikacji), bądź jedynie ustalić tożsamość stron (atak na prywatność). Konstrukcja protokołów AKE składa się zazwyczaj z dwóch faz: fazy wymiany kluczy Diffiego-Hellmana, oraz fazy identyfikacji stron. Protokół zakończony sukcesem dla każdej ze stron oznacza, że: klucz sesyjny został wyliczony, druga strona również wyliczyła ten sam klucz, druga strona została poprawnie zidentyfikowana - czyli jest tą, za którą się podaje. W wyniku tego obie strony przechodzą na komunikację szyfrowaną ustalonym kluczem. Ponadto strony powinny mieć pewność, że klucz ten nie jest dostępny dla osób trzecich, a tym samym szyfrowane nim wiadomości są niedostępne dla osób niepowołanych.

**Zaprzeczalne protokoły AKE dla dokumentów elektronicznych** W pracy [A10] zaproponowano ulepszenie niemieckiego protokołu PACE|AA [1] do identyfikacji i ustalania kluczy sesyjnych pomiędzy stroną nazywaną Alicją ($A$), oznaczającą elektroniczny dokumentem tożsamości - realizowany na karcie mikroprocesorowej, a stroną nazywaną Bobem ($B$), będącą czytnikiem elektronicznego dokumentu. W ramach protokołu karta i czytnik dzielą wspólne krótkie hasło $\pi$. Hasło to karta ma zapisane w pamięci wewnętrznej, a do czytnika jest ono wprowadzane, np. ręcznie przez użytkownika, tuż przed rozpoczęciem sesji. Dodatkowo karta posiada długoterminowy klucz prywatny $x_A$ i odpowiadający mu klucz publiczny $X_A = g^{x_A}$ z certyfikatem $cert_A$. W pierwszej fazie protokołu oba urządzenia wyliczają osobno klucz $K_\pi$ symetrycznego schematu szyfrowania $(ENC, DEC)$. Karta wybiera losową wartość $s \in \mathbb{Z}_q^*$, wysyła do czytnika szyfrogram $z = ENC(K_\pi, s)$ oraz parametry przestrzeni obliczeń $\mathcal{G}$, a czytnik deszyfruje wartość $s = DEC(K_\pi, z)$. W drugiej fazie następuje wstępne wykonanie protokołu Diffiego-Hellmana. Karta i czytnik losują odpowiednio wartości $y_A$ i $y_B$, wysyłają do siebie wzajemnie $Y_A = g^{y_A}$, $Y_B = g^{y_B}$ oraz liczą $h = (Y_A)^{y_B} = (Y_B)^{y_A}$. W trzeciej fazie urządzenia wyliczają nowy generator $\hat{g} = hg^s$ i następuje drugie wykonanie protokołu Diffiego-Hellmana dla tego generatora. Karta i czytnik losują odpowiednio własne wartości $y'_A$ i $y'_B$, wysyłają do siebie wzajemnie $Y'_A = \hat{g}^{y'_A}$ i $Y'_B = \hat{g}^{y'_B}$, oraz liczą niezależnie $K = (Y'_A)^{y'_B} = (Y'_B)^{y'_A}$. Następnie urządzenia wyliczają klucze $K_{ENC} = H(1||K)$, $K'_{SC} = H(2||K)$, $K_{MAC} = H(3||K)$, $K'_{MAC} = H(4||K)$ oraz przesyłają sobie wzajemnie wiadomości uwierzytelniające transmisję: $T_A = MAC(K'_{MAC}, (Y'_B, \mathcal{G}))$, $T_B = MAC(K'_{MAC}, (Y'_A, \mathcal{G}))$. Następnie zostaje przeprowadzana identyfikacja dokumentu. W oryginalnym niemieckim protokole, w jego zaprzeczalnej wersji, karta wylicza $s = y_A + H(Y'_A, (G))x_A$ i wysyła szyfrogram $ENC(K'_{SC}, (s, cert_A))$. Czytnik deszyfruje wartość $s$ i akceptuje tożsamość gdy $g^s = Y_A X_A^{H(Y'_A,(G))}$. W podstawowej modyfikacji zaproponowanej dla fazy identyfikacyjnej dokumentu, karta wylicza wartość $w = y_A/x_A$ wysyła szyfrogram $ENC(K'_{SC},(w,cert_A))$, a czytnik akceptuje tożsamość jeśli $X_A^w = Y_A$. Modyfikacja ta jest niezauważalna z punktu widzenia zewnętrznego obserwatora: nie zmienia liczby komunikatów oraz typów wartości przesyłanych. Dodatkowo modyfikacja nie zmienia właściwości zaprzeczalności protokołu oryginalnego. Zwróćmy uwagę, że protokół oryginalny i podstawowa modyfikacja nie są odporne na wyciek klucza efemerycznego $y_A$. W obu protokołach możliwe jest wtedy obliczenie klucza prywatnego karty przez czytnik. Dlatego w pracy zaproponowano inną modyfikację. W drugiej fazie, podczas wstępnego wykonania protokołu Diffiego-Hellmana, karta wysyła $Y_A = g^{x_A y_A}$, a w końcowej części identyfikacyjnej podstawia $w = y_A$, wysyła szyfrogram $ENC(K'_{SC},(w,cert_A))$, a czytnik akceptuje tożsamość jeśli $X_A^w = Y_A$. Wersja ta również nie zmienia formatu komunikatów i jest zaprzeczalna. Dodatkową cechą jest odporność na wyciek wartości efemerycznych: w żadnym komunikacie tajność klucza prywatnego $x_A$ nie jest zależna od tajności wartości efemerycznej $y_A$.

W pracach [A11] oraz [A12] zaproponowano protokół AKE dla dokumentów elektronicznych. Konstrukcja oby protokołów jest bardzo podobna. Strony protokołu identyfikowane są odpowiednio jako: Alicja posiadająca klucz prywatny $x_A$, klucz publiczny $y_A = g^{x_A}$, certyfikowany certyfikatem $cert_A$; oraz Bob posiadający klucz prywatny $x_B$, klucz publiczny $y_B = g^{x_B}$, certyfikowany certyfikatem $cert_B$.

W pracy [A11] w fazie wymiany klucza Diffiego-Hellmana Alicja generuje losowo klucz efemeryczny $a$ wylicza $h_A = H(a)$ oraz $c_A = g^{h_A}$ i wysyła $c_A$ do Boba. Bob analogicznie generuje losowo klucz efemeryczny $b$ wylicza $h_B = H(b)$ oraz $c_B = g^{h_B}$ i wysyła $c_B$ do Alicji. Następnie strony wyliczają klucz pośredni $K$: Alicja wylicza $K = c_B^{h_A}$, a Bob $K = c_A^{h_B}$; po czym wyprowadzają: $K_A = H(K,1)$, $K_B = H(K,2)$, $K'_A = H(K,3)$, $K'_B = H(K,4)$. Alicja wylicza $r_A = H(c_B^{x_A}, K'_A)$ i wysyła do Boba szyfrogram $Enc_{K_A}(cert_A, r_A)$ zakodowany kluczem $K_A$. Bob odkodowuje $r_A$, sprawdza czy jest równe wartości $H(y_A^{h_B}, K'_A)$. Jeśli tak to liczy $r_B = H(c_A^{x_B}, K'_B)$ i wysyła do Alicji szyfrogram $Enc_{K_B}(cert_B, r_B)$ zakodowany kluczem $K_B$. Alicja odkodowuje $r_B$, sprawdza czy jest równe wartości $H(y_B^{h_A}, K'_B)$. Na koniec obie strony obliczają klucz sesyjny $K_{session} = H(K,5)$. Zauważmy, że warunki sprawdzane przez użytkowników po obu stronach są prawdziwe jedynie gdy obliczenia, wykonywane z udziałem klucza prywatnego po stronie wysyłającej, są równe obliczeniom z udziałem odpowiadającego klucza publicznych po stronie weryfikującej. Szyfrowana wymiana certyfikatów w komunikacie trzecim i czwartym zapewnia prywatność użytkowników - adwersarz nie poznaje ich tożsamości. Protokół jest zaprzeczalny, ponieważ każda ze stron może wygenerować transkrypt protokołu samodzielnie - bez znajomości klucza prywatnego strony drugiej, a wartości $r_A$ oraz $r_B$ wyliczane niezależnie po obu stronach mają taką samą wartość. Bezpieczeństwo klucza sesyjnego zależy od tajności kluczy efemerycznych ustalanych po każdej stronie. Wyciek wartości $a$ lub $b$ kompromituje klucz sesyjny, oraz zdradza tożsamości uczestników protokołu.

W pracy [A12] w fazie wymiany klucza Diffiego-Hellmana Alicja generuje losowo klucz efemeryczny $a$ wylicza $h_A = H(a|0)$ oraz $c_A = y_A^{H(a|0)}$, i wysyła $c_A$ do Boba. Bob analogicznie generuje losowo klucz efemeryczny $b$ wylicza $h_B = H(b|0)$ oraz $c_B = y_B^{H(b|0)}$, i wysyła $c_B$ do Alicji. Następnie wyliczają klucz pośredni $K$: Alicja $K = c_B^{x_A h_A}$, a Bob $K = c_A^{x_B h_B}$. Następnie obie strony wyznaczają: $K_A = H(K, 1)$, $K_B = H(K, 2)$. Alicja wysyła do Boba szyfrogram $Enc_{K_A}(a, cert_A)$. Bob wylicza $a$ oraz $cert_A$ i sprawdza czy $c_A$ otrzymane w pierwszym komunikacie jest równe $y_A^{H(a|0)}$ dla $y_A$ uzyskanego z certyfikatu $cert_A$. Jeśli tak to wysyła do Alicji szyfrogram $Enc_{K_B}(b, cert_B)$. Alicja wylicza $b$ oraz $cert_B$ i sprawdza czy $c_B$ otrzymane w drugim komunikacie protokołu jest równe $y_B^{H(b|0)}$ dla $y_B$ uzyskanego z certyfikatu $cert_B$. Na koniec obie strony liczą klucz sesyjny $K_s = H(K, 3)$. Szyfrowana wymiana certyfikatów w komunikacie trzecim i czwartym zapewnia prywatność użytkowników - adwersarz nie poznaje ich tożsamości. Protokół jest zaprzeczalny, ponieważ każda ze stron może wygenerować transkrypt protokołu samodzielnie - bez znajomości klucza prywatnego strony drugiej, a wartość klucza tymczasowego $K = g^{x_A H(a|0) x_B H(b|0)}$ wyliczana niezależnie po obu stronach ma taką samą wartość. Bezpieczeństwo klucza sesyjnego jest silniejsze niż w poprzednim protokole, zależy od tajności kluczy efemerycznych ustalanych po każdej stronie oraz od tajności długoterminowych kluczy prywatnych. Wyciek samych wartości efemerycznych $a$ lub $b$ nie kompromituje klucza tymczasowego, i w rezultacie sesyjnego. Kompromitacja następuje jedynie w przypadku jednoczesnego wycieku obu kluczy (efemerycznego i długoterminowego) jednej ze stron.

**Zaprzeczalna wersja protokołu SIGMA** W pracy [A13] zaproponowałem zaprzeczalną wersję protokołu SIGMA [38, 39]. Protokół SIGMA jest protokołem autoryzacji i wymiany klucza sesyjnego zbudowanym w oparciu o bezpieczne schematy kryptograficzne: funkcję podpisu $\textsc{sig}$, funkcję pseudolosową $\textsc{prf}$, oraz kod uwierzytelnienia wiadomości $\mathsf{MAC}$. Użytkownicy protokołu, inicjator i respondent, oznaczeni odpowiednio literami $I$, $R$, wykonują obliczenia w ustalonej grupie $G$, w której problem CDH jest trudny. $I$, $R$ posługują się parami kluczy prywatny/publiczny wybranego schematu podpisu $\textsc{sig}$ w zdefiniowanego grupie $G$, odpowiednio: $(\mathsf{sk}_I, \mathsf{pk}_I)$ i $(\mathsf{sk}_R, \mathsf{pk}_R)$. Protokół składa się z dwóch faz. W pierwszej fazie następuje uzgodnienie klucza pośredniego metodą Diffiego-Hellmana. W drugiej fazie następuje identyfikacja i uwierzytelnienie stron za pomocą wybranego schematu podpisu oraz kodu uwierzytelnienia wiadomości.

Trzyrundowa wersja tego protokołu, w której respondent jako pierwszy dowodzi swojej tożsamości, ma następujące kroki. W pierwszym kroku $I$ wybiera losowo identyfikator sesji $s$, oraz klucz efemeryczny $x$ i wysyła parę $s, X = g^x$ do $R$. W drugim kroku $R$ wybiera losowo klucz efemeryczny $y$, oblicza klucz $k_1 = \textsc{prf}_{g^{xy}}(1)$ funkcji $\mathsf{MAC}$, oraz klucz sesyjny $k_0 = \textsc{prf}_{g^{xy}}(0)$. Wysyła do $I$ wartości $s, X = g^y$, własny certyfikat $ID_R$, dowód własnej tożsamości w formie podpisu $\textsc{sig}_{\mathsf{sk}_R}(”1”, s, g^x, g^y)$, oraz uwierzytelnienie $\mathsf{MAC}_{k_1}(”1”, s, ID_R)$. W trzecim kroku $I$ wylicza $k_1 = \textsc{prf}_{g^{xy}}(1)$, weryfikuje uwierzytelnienie $\mathsf{MAC}_{k_1}(”1”, s, ID_R)$, weryfikuje podpis $\textsc{sig}_{\mathsf{sk}_R}(”1”, s, g^x, g^y)$ na podstawie klucza publicznego związanego z $ID_R$, i jeśli weryfikacje są poprawne to oblicza klucz sesyjny $k_0 = \textsc{prf}_{g^{xy}}(0)$, a następnie wysyła do $R$ własny certyfikat $ID_I$, dowód własnej tożsamości w formie podpisu $\textsc{sig}_{\mathsf{sk}_I}(”0”, s, g^x, g^y)$, oraz uwierzytelnienie $\mathsf{MAC}_{k_1}(”0”, s, ID_I)$. W czwartym kroku $R$ weryfikuje uwierzytelnienie $\mathsf{MAC}_{k_1}(”0”, s, ID_I)$, weryfikuje podpis $\textsc{sig}_{\mathsf{sk}_I}(”0”, s, g^x, g^y)$ na podstawie klucza publicznego związanego z $ID_R$. W czterorundowej wersji tego protokołu to inicjator pierwszy dowodzi swojej tożsamości: w drugim kroku $R$ wysyła do $I$ jedynie $s, X = g^y$, a informacje uwierzytelniające $\mathsf{MAC}_{k_1}(''1'', s, ID_R)$ oraz $\textsc{sig}_{\mathsf{sk}_R}(''1'', s, g^x, g^y)$, wysyłane są do $I$ w ostatniej czwartej wiadomości protokołu.

Zwróćmy uwagę, że protokół w wersji oryginalnej jest niezaprzeczalny. Strony protokołu podpisując wymieniane wiadomości własnymi kluczami tajnymi, nie mogą się wyprzeć swojego udziału w sesji protokołu. W związku z powyższym w omawianej pracy zaproponowano silniejszą, w stosunku do poprzednich notacji [40, 41] definicję zaprzeczalności. W kontekście protokołów wymiany klucza protokół jest zaprzeczalny dla jednej ze stron jeśli transkrypt sesji, zawierający wszystkie komunikaty sesji, nie stanowi dowodu uczestnictwa tej strony w sesji, nawet gdy ujawnione są klucze sekretne stron - służące do składania podpisów. W omawianej pracy zaproponowałem zastosowanie w protokole podpisów pierścieniowych. Podczas tworzenia podpisu podpisujący, oprócz swojego klucza prywatnego, używa również kluczy publicznych innych użytkowników, w taki sposób, że weryfikujący nie ma możliwości stwierdzenia, która z uwikłanych osób rzeczywiście podpis skonstruowała. W tym kontekście modyfikacja polega na wykorzystaniu podpisów $\textsc{rsig}_{\mathsf{sk}_I,\{\mathsf{pk}_I,\mathsf{pk}_R\}}()$ oraz $\textsc{rsig}_{\mathsf{sk}_R,\{\mathsf{pk}_I,\mathsf{pk}_R\}}()$ odpowiednio przez $I$ oraz $R$, używających własnych kluczy prywatnych $\mathsf{sk}_I, \mathsf{sk}_R$, i publicznych kluczy $\mathsf{pk}_I, \mathsf{pk}_R$. W ten użytkownicy wciągają do pierścienia wzajemnie drugą stronę protokołu, z którą ustanawiają klucz sesyjny. Anonimowość podpisów pierścieniowych sprawia, że dla zewnętrznego obserwatora nie ma możliwości zweryfikowania, który z uczestników pierścienia jest rzeczywistym podpisującym. Transkrypt protokołu wymiany klucza SIGMA z wykorzystaniem podpisów pierścieniowych jest nierozróżnialny od transkryptu tego protokołu wygenerowanego przez jedną ze stron samodzielnie. W pracy szczegółowo zaproponowano dwie zaprzeczalne wersje protokołu SIGMA: trzyrundową, zaprzeczalną dla strony inicjującej protokół i czterorundową zaprzeczalną dla strony odpowiadającej protokołu.

Dodatkowym rozszerzeniem zaproponowanym przeze mnie w pracy [A13] jest wzmocnienie bezpieczeństwa klucza sesyjnego. W przypadku gdy klucze długoterminowe zdefiniowane są w tej samej grupie co klucze efemeryczne, np. gdy zastosujemy schemat podpisów pierścieniowych [42], możliwe staje się zastąpienie efemerycznych kluczy publicznych $g^x, g^y$, kluczami $g^{\mathsf{sk}_I x}, g^{\mathsf{sk}_R y}$. Klucz pośredni wyliczany przez każdą ze stron ma wtedy postać $g^{\mathsf{sk}_I x \mathsf{sk}_R y}$ (zamiast $g^{xy}$). Dzięki temu protokół staje się odporny na ataki kompromitujące jedynie klucze efemeryczne $x$, $y$. Skuteczny atak może nastąpić jedynie po poznaniu przez adwersarza obu kluczy jednej ze stron: efemerycznego i długoterminowego. Omawiana praca [A13] jest rozszerzoną wersją pracy konferencyjnej [4], wyłącznie mojego autorstwa. W stosunku do wersji konferencyjnej, w rozszerzonej wersji dodatkowo: 1) rozwinięto analizę bezpieczeństwa i zaprzeczalności schematu; 2) wykonano prototypowe implementacje protokołów w językach *Python* (*Charm Crypto Library*) i *Java*, oraz przeprowadzono testy tych dla prototypów; 3) uwzględniono analizę implementowalnosci proponowanych protokołów na urządzeniach małej mocy z oszacowaniem złożoności energetycznej i czasowej.

**Zaprzeczalna wersja protokołu HMQV odporna na ataki eKCI** W pracy [A14] zaproponowałem zaprzeczalną wersję protokołu HMQV [6] odporną na rozszerzony atak personifikacji z kluczem tajnym i kluczem efemerycznym (ang. Extended Key Compromise Impersonation -

eKCI). Atak ten, zaproponowany w [7] polega na tym, że adwersarz posiadający dostęp do kluczy efemerycznych i długoterminowych jednej ze stron może przeprowadzić z sukcesem protokół ustalania kluczy sesyjnych z tą stroną, przybierając tożsamość dowolnego innego użytkownika zarejestrowanego w systemie. Jako przykład protokołu wrażliwego na ten atak, autorzy pracy [7] podają protokół HMQV. W ramach tego protokołu użytkownicy Alicja i Bob wykonują obliczenia w ustalonej grupie $G$, w której problem CDH jest trudny. Alicja i Bob posługują się parami kluczy prywatny/publiczny, odpowiednio: $(a, g^a)$ i $(b, g^b)$, a także odpowiednio ustalonymi funkcjami skrótu $\mathsf{H}, \bar{\mathsf{H}}$, oraz funkcją do uwierzytelniania wiadomości $\mathsf{MAC}$. W pierwszym kroku Alicja wybiera losowo klucz efemeryczny $x \leftarrow_{\$} \{0, \ldots, org(G) - 1\}$ i wysyła $X = g^x$ do Boba. Bob wybiera losowo swój klucz efemeryczny $y \leftarrow_{\$} \{0, \ldots, org(G) - 1\}$, oblicza $d = \bar{\mathsf{H}}(X || \text{``}Bob\text{''})$, $e = \bar{\mathsf{H}}(Y || \text{``}Alice\text{''})$, $\sigma_b = (Xg^{ad})^{y+eb}$, $k_m = \mathsf{H}(\sigma_b || 0)$, $Z = \mathsf{MAC}(\text{''}1\text{''}, k_m)$, a następnie wysyła $Y = g^y$ oraz $Z$ do Alicji. Alicja wylicza $\sigma_a = (Yg^{be})^{x+da}$ dla $d$, $e$ wyliczonych analogicznie jak przez Boba, $k_m = \mathsf{H}(\sigma_a || 0)$. Następnie weryfikuje czy $Z = \mathsf{MAC}(\text{''}1\text{''}, k_m)$ i jeśli tak jest to wysyła $W = \mathsf{MAC}(\text{''}0\text{''}, k_m)$ do Boba. Bob weryfikuje czy $W = \mathsf{MAC}(\text{''}0\text{''}, k_m)$. Obie strony wyliczają klucz sesyjny odpowiednio jako: $\mathsf{H}(\sigma_a || 1)$ i $\mathsf{H}(\sigma_b || 1)$. Zwróćmy uwagę, że $\sigma_a = (Yg^{be})^{x+da} = (g^y g^{be})^{x+da} = g^{(x+da)(y+eb)} = (g^x g^{da})^{y+eb} = (Xg^{da})^{y+eb} = \sigma_b$. Dzięki temu obie strony mogą wyliczyć niezależnie $k_m$ oraz wartości $Z$ i $W$. Ponadto klucz sesyjny wyliczony po obu stronach jest równy. Z tego właśnie powodu protokół jest wrażliwy na atak eKCI, gdyż strony nie udowadniają sobie wzajemnie znajomości swoich kluczy tajnych, a jedynie domniemają poprawności deklarowanych tożsamości na podstawie wyliczenia tych samych wartości kluczy sesyjnych. Autorzy [7] proponują uzupełnienie protokołu o dodatkową warstwą uwierzytelniającą wykorzystującą podpisy BLS [43]. Bob i Alicja wysyłają sobie wzajemnie dodatkowo podpisy postaci $V = (\mathsf{H}_1(m))^b, V' = (\mathsf{H}_1(m))^a$ weryfikowane za pomocą kluczy publicznych $B = g^b, A = g^a$, dla ustalonej funkcji skrótu $\mathsf{H}_1$ i wiadomości $m$=”Alice”||”Bob”$||A||B||X||Y$. Zauważmy jednak, że takie podejście powoduje, że protokół ten przestaje być zaprzeczalny: żadna ze stron nie może wygenerować jego transkryptu samodzielnie. Wobec tego w pracy [A14] zaproponowano zastąpienie podpisów BLS w warstwie uwierzytelniania komunikatami zmodyfikowanego schematu Schnorra z pracy [A5]. Bob i Alicja wysyłają sobie wzajemnie dowody tożsamości odpowiednio: $S = (\mathsf{H}_1(X))^x (\mathsf{H}_1(X))^{ac}$ oraz $V = (\mathsf{H}_1(Y))^y (\mathsf{H}_1(Y))^{bc}$, dla $m_A$ = ”Alice”||”Bob”$||A||B||Y$ i wyzwania $c = \mathsf{H}_2(m_A)$ liczonego po stronie Alicji, oraz $m_B$ = ”Alice”||”Bob”$||A||B||X$ i wyzwania $c = \mathsf{H}_2(m_B)$ liczonego po stronie Boba. Wartości te weryfikowane są zgodnie ze schematem [A5], opisanym w punkcie 4.3.4. Wykorzystanie schematu [A5] gwarantuje bezpieczeństwo kluczy długoterminowych w przypadku kompromitacji jedynie kluczy efemerycznych, a dodatkowo przywraca zaprzeczalność oryginalnego protokołu HMQV. W pracy przeprowadzono analizę bezpieczeństwa kluczy sesyjnych proponowanych schematów (redukcja do problemu CDH), analizę zaprzeczalności (symulowalność dla jednej ze stron), oraz zaprezentowano wyniki uzyskane dla testowej implementacji porównawczej omawianych protokołów.

**Protokoły uwierzytelnionego ustalania kluczy dla pojazdów** Poniżej omawiamy pokrótce wyniki uzyskane dla protokołów uwierzytelniania pojazdów. Mogą być one rozpatrywane jako praktyczne rozszerzenie rezultatów omówionych w podrozdziałach 4.3.4 i 4.3.5. W odróżnieniu od typowych protokołów uwierzytelniania i wymiany klucza, w których użytkownicy identyfikują się i komunikują w sposób zdalny (często na duże odległości, bez identyfikacji optycznej), w protokołach przeznaczonych dla pojazdów identyfikacja optyczna może mieć duże znaczenie. Samochody mogą w sposób autonomiczny wymieniać komunikaty, nawiązując łączność bezprzewodową w kanale radiowym i identyfikując się certyfikowanymi kluczami. Jednak dla kierowcy pojazdu, rozpoznającego inne samochody w zasięgu wzroku po atrybutach związanych z wyglądem (marka, model, kolor), identyfikacja ta może nie być jednoznaczna, co z kolei może prowadzić do niebezpiecznych zdarzeń drogowych. W tym kontekście funkcjonalność tych protokołów obejmuje przypadki, w których wymiana kluczy sesyjnych powinna nastąpić nie tylko po pozytywnej weryfikacji za pomocą certyfikowanych kluczy liczbowych, ale również po pozytywnej weryfikacji atrybutów fizycznych pojazdu. W przytaczanych pracach zakłada się, że pojazdy wyposażone są w odpowiednie urządzenia: kamery, sensory optyczne, itp. umożliwiające rozpoznawanie dodatkowych atrybutów fizycznych.

W pracy [A15] zaproponowano metodę autoryzacji i ustalania klucza sesyjnego dla pojazdów, których wygląd jest certyfikowany wraz z kluczem publicznym. Certyfikat taki wiązałby następujące atrybuty: identyfikator producenta pojazdu, opis atrybutów wyglądu identyfikowanych optycznie (np. marka, model, rok, logo, rodzaj nadwozia, kolor), sekcje identyfikatorów zwykłych (np. nr silnika, nr nadwozia), procedury wymagane przy weryfikacji z kluczem publicznym dla wybranego schematu kryptograficznego. Przyjęto następujące oznaczenia: $Cert$ oznacza certyfikat, $SN$ jest numerem certyfikatu, $Attribute$ jest zbiorem certyfikowanych statycznych zewnętrznych atrybutów, $CA$ jest urzędem certyfikacyjnym, $Sign$ jest algorytmem ustalonego schematu podpisu, gdzie $(PK, SK)$ oznacza parę (klucz publiczny, klucz prywatny), a $E, D$ są operacjami szyfrowania i deszyfrowania wybranego asymetrycznego schematu szyfrowania, w którym pary klucz publiczny i prywatny oznaczono przez $(PKey, SKey)$. Dla tego schematu zakładamy, że zachodzi: $D_{SKey}(E_{PKey}(m)) = m$ oraz $D_{PKey}(E_{SKey}(m)) = m.$ Dla dalszych rozważań w pracy przedstawiono prosty dwurundowy schemat wykorzystujący proponowany typ certyfikatu. Nadawca $S$ wysyła do odbiorcy $R$ certyfikat postaci $Cert_S = Attribute_S + PKey_S || Sign_{CA}(H(Attribute_S + PKey_S))$. Odbiorca $R$ weryfikuje certyfikat $Cert_S$, podpis $Sign_{CA}()$, statyczne zewnętrzne atrybuty $Attribute_S$. Jeśli weryfikacja jest poprawna to $R$ wysyła do $S$ certyfikat $Cert_R = Attribute_R + PKey_R || Sign_{CA}(H(Attribute_R + PKey_R))$, oraz szyfrogramy: $E_{PKey_S}(key_r + SN_S) || E_{PKey_S}(E_{SKey_R}(H(key_r + SN_S)))$. Nadawca $S$ weryfikuje certyfikat $Cert_R$, podpis $Sign_{CA}()$ oraz statyczne zewnętrzne atrybuty $Attribute_R$. Deszyfruje szyfrogramy od $R$. Szyfrogram $E_{PKey_S}(E_{SKey_R}(H(key_r + SN_S)))$ pełni role uwierzytelniającą dla wiadomości szyfrowanej w $E_{PKey_S}(key_r + SN_S)$. Obie strony przyjmują wartość $key_r$ jako klucz sesyjny.

W pracy zauważono i przeanalizowano brak odporności powyższego schematu na pewne ataki powtórzeniowe adwersarza $\mathcal{A}$, który znajduje się w zasięgu nadawcy $S$ i odbiorcy $R$, i który rejestruje komunikacje pomiędzy tymi pojazdami. W pierwszym scenariuszu adwersarz $\mathcal{A}$ podaje fałszywą tożsamość $S$: inicjuje komunikację od $S$ do $R$ poprzez odtwarzanie pierwszego komunikatu wcześniej zarejestrowanego protokołu, oraz komunikatów następujących po odpowiedzi od $R$. Pomimo tego, że nie potrafi odkodować przekazu może jednak sprawić, że $R$ będzie przekonany, że rozmawia z $S$. W drugim scenariuszu adwersarz $\mathcal{A}$, kiedy odbierze wywołanie od $S$, może podawać się fałszywie za $R$, odsyłając do $S$ wcześniej zarejestrowany drugi komunikat protokołu. Dodatkowo zwrócono uwagę na fakt, że bezpieczeństwo kluczy sesyjnych zależy jedynie od tajności kluczy długoterminowych. Ujawnienie kluczy deszyfrujących w przyszłości kompromituje klucze sesyjne ustalone w przeszłości. Najistotniejsza z proponowanych modyfikacji polega na użyciu efemerycznych kluczy Diffiego-Hellmana w celu ustalenia klucza sesyjnego zależnego od obu stron. W tym przypadku w pierwszym kroku $S$ wysyła do $R$ wiadomość $Cert_S|Nounce_S$, gdzie $Nounce_S = g^{\alpha}$. W drugiej wiadomości wartość $key_r$ jest ustalana przez $R$ jako $Nounce_R = g^{\beta}$. Klucz sesyjny po oby stronach wyliczony jest jako wartość $g^{\alpha\beta}$. W przypadku gdy efemeryczne wartości $\alpha, \beta$ kasowane są przez strony każdorazowo po ustaleniu kluczy sesyjnych, adwersarz posiadający klucze długoterminowe musi rozwiązać problem $CDH(g, g^{\alpha}, g^{\beta})$ aby uzyskać klucz sesyjny. W pracy wskazano dodatkowo na możliwe wykorzystanie proponowanego certyfikatu wraz z bezpiecznymi trzyrundowymi protokołami AKE takimi jak ISO-KE [44] lub SIGMA [38, 39, 5]. W trakcie wykonywania wybranego protokołu pozytywna weryfikacji certyfikatu polegałaby na dodatkowym sprawdzeniu atrybutów wyglądu.

Protokół zaproponowany w pracy [A16] jest rozszerzeniem protokołu [A15]. Główną różnicą jest propozycja dodatkowego wykorzystania laserowego kanału transmisji dla wysyłania i odbierania certyfikatów wymienianych pomiędzy pojazdami. Kierunkowa i spójna charakterystyka tego kanału zapewnia jednoznaczność w przypadku próby nawiązania komunikacji i wizualnej identyfikacji z pojazdami wyglądającymi identycznie (ten sam model, marka, kolor). Zakłada się, że pojazdy wyposażone są w odpowiednie urządzenia nadawczo-odbiorcze: czytnik wiązki laserowej u odbiorcy, oraz laser modulujący sygnał komunikatów po stronie nadawcy są umiejscowiony w taki sposób, że możliwe jest efektywne pozycjonowanie wiązki nadawczej na czytniku odbiorcy, i utrzymywanie połączenia pomimo dynamicznej zmiana położenia oraz prędkości obu samochodów. Raz zainicjowana wiązka laserowa podąża za pojazdem. Umiejscowienie wiązki na niewłaściwym pojeździe jest równoważne z nieotrzymaniem komunikatu przez pojazd odbiorcy i prowadzi do przerwania wykonywania protokołu.

W pracy [A17] rozwinięto propozycje z prac [A15, A16]. Zaproponowany protokół działa w dwóch warstwach fizycznych: w kanale radiowym, w którym wykonywany jest typowy protokół AKE do ustalania kluczy sesyjnych, oraz w kanale optycznym, w którym do identyfikacji i śledzenia pojazdów wykorzystywane są funkcje fizycznie nieklonowalne (ang. Physically Unclonable Functions - PUF). Proponowane konstrukcja pozwala uzyskać protokół odporny na ataki koalicji adwersarzy typu "man-in-the-middle"przekierowujących komunikaty do innych pojazdów pośredniczących.

Wykorzystanie urządzeń PUF wymaga specjalnego przygotowania i wiąże się z instalacją odpowiednich urządzeń: urządzenia wysyłającego modulowane wyzwania pobudzające, oraz urządzenia odczytującego wyzwania. Zbieranie kolekcji wyzwań i odpowiedzi, tzw. par CRP (ang. Challenge-Response Pair), następuje w fazie uczenia w bezpieczny sposób. Optyczny PUF może mieć postać karty z przeźroczystą płytką. Płytka ta wykonana jest z materiału zawierającego mikrocząsteczki lub mikrouszkodzenia uzyskane w sposób losowy na etapie produkcji, np. wskutek zanieczyszczania szkła, lub napylania powierzchni drobinami ściernymi. Płytka jest umieszczana w czytniku, a następnie prześwietlana wiązką laserową modulowaną zgodnie z $i$-tym wyzwaniem $c_i$. Uzyskany obraz interferencyjny $s_i$ jest rejestrowany za pomocą specjalnych fotodiod i dekodowany do postaci numerycznej $r_i$. Intuicyjny schemat identyfikacji podzielony jest na dwa etapy. W fazie kolekcjonowania przez urządzenie PUF przepuszcza się wektor wyzwań $\mathrm{C} = (c_1, c_2, ...c_i, ...c_n)$ uzyskując wektor odpowiedzi $\mathrm{R} = (r_1, r_2, ...r_i, ...r_n)$. W fazie identyfikacji urządzenie PUF jest stymulowane wiązką laserową modulowaną wyzwaniem $c_i$. Akceptacja następuje, jeśli uzyskana odpowiedź $r_i$ jest równa wartości uzyskanej na etapie uczenia. Powyższy schemat jest adaptowany na potrzeby pojazdów w następujący sposób. Urządzenie PUF, wraz z wymaganą optyką, jest montowane w pojeździe jako identyfikator. Laser wraz optyką jest montowany jako część służąca do zdalnego stymulowania urządzeń PUF innych pojazdów. Pojazd testujący generuje modulowaną $i$tą wiązkę świetlną $m_i = f(c_i)$ za pomocą sprzętowo zaimplementowanej deterministycznej funkcji $f$, przyjmującej na wejściu wartości numeryczne $c_i$. Wiązka $m_i$ pobudza płytkę $\wp$ urządzenia PUF pojazdu testowanego, tworząc unikalny obraz interferencyjny $s_i$=$\wp(m_i)$, który z kolei jest dekodowany w pojeździe testującym do wartości numerycznej $r_i = w(s_i)$ za pomocą sprzętowo zaimplementowanej deterministycznej funkcji $w$. Każdy pojazd przechowuje w bezpiecznej i nieulotnej pamięci certyfikaty dla par CRP, uzyskane za pomocą własnego urządzenia PUF. Certyfikaty $Cert(c_{i,\hat{A}}, r_{i,\hat{A}}, Attribute_{\hat{A}}, \hat{A}, A, t_{val})$ dla każdej pary wiążą odpowiednio: wartości CRP, atrybuty statyczne, takie jak nr rejestracyjny, marka, model, identyfikator pojazdu (np. nr VIN), klucz publiczny dla kanału radiowego, okres ważności. Certyfikaty są jednorazowe i ważne jedynie w okresie zdefiniowanym przez czas $t_{val}$. Zakładamy dodatkowo, że skuteczny atak adwersarza na urządzenie PUF, polegający na wygenerowaniu właściwej odpowiedzi dla poznanego wyzwania wymaga odpowiednio długiego okresu czasu ($t_{adv}$). Zastosowane podejście pozwala związać dowolny protokół uzgadniania klucza w warstwie komunikacji radiowej z proponowanym protokołem identyfikacji optycznej wykonywanym za pomocą urządzeń PUF.

W pracy, bez straty ogólności, przedstawiono wiązanie dla protokołu CMQV [45]. Jest ono następujące: Zakładamy, że pojazdy oznaczone jako $\hat{A}$ i $\hat{B}$ posiadają odpowiednio pary kluczy prywatnych/publicznych $(a, A = g^a)$ i $(b, B = g^b)$ w grupie generowane przez $g$. Ponadto korzystają z odpowiednio ustalonych funkcji skrótu $\mathsf{H}, \mathsf{H}_1, \mathsf{H}_2$. Pojazd inicjujący $\hat{A}$ wyznacza tajny klucz efemeryczny $x \leftarrow_{\$} \{0,1\}^{\lambda}$, oblicza publiczny klucz efemeryczny $X = g^{\mathsf{H}_1(x,a)}$, tworzy identyfikator sesji $s = (I, \hat{A}, \hat{B}, X, *)$ gdzie $I$ oznacza rolę, a $*$ oznacza wymagany publiczny klucz efemeryczny strony odpowiadającej wypełniany w trakcie wykonywania protokołu. $\hat{A}$ przełącza się w tryb radiowy i wysyła swój niewykorzystany certyfikat $Cert(c_{i,\hat{A}}, r_{i,\hat{A}}, Attribute_{\hat{A}}, \hat{A}, A, t_{val})$ oraz identyfikator sesji $s = (I, \hat{A}, \hat{B}, X, *)$ do pojazdu $\hat{B}$. $\hat{B}$ weryfikuje certyfikat, weryfikuje atrybuty $Attribute_{\hat{A}}$, wydobywa parę $(c_{i,\hat{A}}, r_{i,\hat{A}})$, w oparciu o $t_{val}$ sprawdza czy nie upłynął jej termin ważności, tzn. czy certyfikat nie jest starszy niż potencjalny czasu ataku $t_{adv}$. W przypadku sukcesu $\hat{B}$ przechodzi do warstwy optycznej: modeluje wyzwanie $f(c_{i,\hat{A}})$, wysyła wiązkę wyzwania $m_{i,\hat{B}} = f(c_{i,\hat{A}})$ do pojazdu $\hat{A}$ i odbiera odpowiedź $s_{i,\hat{A}} = \wp_{\hat{A}}(m_{i,\hat{B}})$, którą dekoduje do wartości $r'_{i,\hat{A}} = w(s_{i,\hat{A}})$. Jeśli $r'_{i,\hat{A}}$ jest równe wartości $r_{i,\hat{A}}$ z certyfikatu, to $\hat{B}$ uznaje, że certyfikat odebrany drogą radiową pochodzi rzeczywiście od pojazdu $\hat{A}$ - wtedy kontynuuje wyliczenie klucza sesyjnego. Wyznacza własny tajny klucz efemeryczny $y \leftarrow_{\$} \{0,1\}^{\lambda}$, oblicza publiczny klucz efemeryczny

$Y = g^{\mathsf{H}_1(y,b)}$, oblicza wartości $E = \mathsf{H}_2(Y, \hat{A}, \hat{B})$, $D = \mathsf{H}_2(X, \hat{A}, \hat{B})$, $\sigma = (XA^D)^{\mathsf{H}_1(y,b)+Eb}$ oraz klucz sesyjny $k = \mathsf{H}(\sigma, X, Y, \hat{A}, \hat{B})$ dla sesji $s = (R, \hat{B}, \hat{A}, X, Y)$. Następnie wysyła do $\hat{A}$ identyfikator sesji $s$ oraz własny niewykorzystany certyfikat $Cert(c_{i,\hat{B}}, r_{i,\hat{B}}, Attribute_{\hat{B}}, \hat{B}, B, t_{val})$. Tym razem pojazd $\hat{A}$ weryfikuje w kanale optycznym pojazd $\hat{B}$ dla sesji $s$ - w sposób analogiczny do weryfikacji wykonanej przez $\hat{B}$. W przypadku sukcesu wylicza $E = \mathsf{H}_2(Y, \hat{A}, \hat{B})$, $D = \mathsf{H}_2(X, \hat{A}, \hat{B})$, oraz $\sigma = (YB^E)^{\mathsf{H}_1(x,a)+Da}$. Zwróćmy uwagę, że wartość ta jest równa wartości wyliczonej niezależnie przez $\hat{B}$. W związku z tym również klucz sesyjny $k = \mathsf{H}(\sigma, X, Y, \hat{A}, \hat{B})$ obliczony przez $\hat{A}$ będzie równy kluczowi wyliczonemu przez $\hat{B}$ w tej sesji.

Zaproponowane w pracach [A15, A16, A17] metody ustalania kluczy są podstawą uzyskania przez autorów patentu w USA [8].

# 5 Omówienie pozostałych osiągnięć naukowo - badawczych

Oprócz prac badawczych, omówionych w poprzednim rozdziale, stanowiących osiągnięcie w rozprawie habilitacyjnej, uczestniczyłem aktywnie w badaniach w innych obszarach.

## 5.1 Podpisy anonimowe

W pracy [B1] zaproponowałem schemat podpisów pierścieniowych, w którym rzeczywisty podpisujący ma możliwość przekazania każdemu użytkownikowi wciągniętemu do pierścienia, tajnego klucza wykorzystywanego do przeprowadzenia dowodu zaprzeczenia autorstwa tworzenia podpisu przez tego użytkownika. Tym sposobem liczba potencjalnych podpisujących może ulec zmniejszeniu, jeśli użytkownicy posiadający odpowiedni klucz, przeprowadzą właściwy nieinteraktywny dowód i dołączą go do oryginalnego podpisu.

W pacy [B2] zaproponowałem rozszerzenie schematu podpisów pierścieniowych o strukturę hierarchiczną pozwalającą na tworzenie podpisów warstwami. Struktura może mieć postać drzewa. W liściach tworzone są podpisy pierścieniowe w oparciu o klucze publiczne użytkowników zarejestrowanych w istniejącej Infrastrukturze Klucza Publicznego (IKP). W innych węzłach tworzone są podpisy pierścieniowe - każdy wraz z nową, wygenerowaną i uwierzytelnioną w tym węźle, parą kluczy. Podpisy w węzłach warstw wyższych mogą być tworzone w oparciu o klucze publiczne dowolnej warstwy niższej. W ten sposób proces tworzenia nowych podpisów jest krótszy, a jego złożoność obliczeniowa jest proporcjonalna do liczby wciągniętych kluczy publicznych z dowolnej warstwy. Natomiast tzw. "zbiór anonimowości" czyli zbiór potencjalnych podpisujących jest znacznie większy i zawiera wszystkie zarejestrowane w IKP klucze publiczne, wykorzystane w liściach poddrzewa wychodzącego z nowego węzła tworzonego tym podpisem. Dzięki tamu, im wyższy jest poziom podpisu, tym "większy" jest zbiór anonimowości podpisującego. Jednocześnie tworzenie nowych podpisów na wyższych poziomach - zapewniających wyższą anonimowość ma niską złożoność. Schemat wprowadza również częściowy porządek na tworzonych podpisach:

podpisy warstw wyższych są późniejsze chronologicznie niż podpisy warstw niższych. Zauważmy, że schematy z prac [B1, B2] mogą być wykorzystane w schemacie proponowanym w pracy [A13], omawianym w punkcie 4.3.5, wprowadzając odpowiednio możliwość potwierdzania udziału w protokole, bądź zwiększenia anonimowości stron biorących udział w protokole.

W pracy [B3] zaproponowałem schemat nietypowych podpisów grupowych, w ramach których pewna grupa podpisujących, o liczności $d$, ze zbioru wszystkich zarejestrowanych użytkowników $\Omega$ (dla $|\Omega| \gg d$), może wspólnie złożyć podpis pod wybranym dokumentem. Weryfikujący nie jest w stanie ustalić tożsamości użytkowników składających podpis. Cechą wyróżniającą jest możliwość dodatkowego potwierdzenia udziału w podpisie przez rzeczywistych podpisujących, oraz zaprzeczenia temu przez użytkowników niepodpisujących. Konstrukcja schematu opiera się na interpolacji Lagranage'a w wykładniku i jest zbliżona do konstrukcji schematu kodowanego rozgłaszania z pracy [A3] omawianego w punkcie 4.3.3. Klucze użytkowników są traktowane jako elementy zbioru interpolacyjnego pewnego wielomianu. Dowolny pełny zbiór interpolacyjny pozwala zrekonstruować wielomian. Natomiast znajomość współczynników wielomianu nie jest wystarczająca do określenia wyjściowego zbioru interpolacyjnego - nie pozwala zatem zidentyfikować użytkowników podpisujących bez znajomości ich udziałów. Olbrzymia liczba kombinacji potencjalnych udziałów, gwarantuje anonimowość nawet w sytuacji, gdy niektórzy użytkownicy wykonają procedurę zaprzeczenia, np. narzuconą im ze względów administracyjnych.

## 5.2 Schematy dla zdalnych platform obliczeniowych

Problem weryfikowania trwałości danych przechowywanych w chmurze obliczeniowej polega na znalezieniu efektywnego protokołu, za pomocą którego użytkownik upewnia się, że dane składowane zdalnie nie zostały zmienione, lub skasowane. Wymaganiami dla schematów stosowanych w tym przypadku jest małą ilość informacji utrzymywana po stronie użytkownika (mały zbiór tzw. metadanych) - zdecydowanie mniejsza niż w przypadku zachowania oryginalnych danych, oraz mała złożoność komunikacyjna, a w szczególności brak konieczności przesyłania oryginalnych plików z chmury do użytkownika jedynie w celach weryfikacji ich spójności.

W pracy [B4] zaproponowałem schemat weryfikowania trwałości danych przechowywanych w chmurze obliczeniowej, oparty na schemacie dzielenia tajemnic i interpolacji Lagrange'a w wykładniku. Schemat swoją konstrukcją nawiązuje do schematów kodowanego rozgłaszania - opisywanych w podrozdziale 4.3.3. Dane przechowywane w chmurze dzielone są na fragmenty interpretowane jako niepełny zbiór interpolacyjny pewnego tajnego wielomianu użytkownika. Sprawdzenie trwałości danych polega na wysłaniu do chmury wyzwania zawierającego, brakujące udziały w wykładniku zamaskowanym wartością losową. Aby poprawnie zinterpolować nowy wielomian zdalny serwer musi wykorzystać wszystkie składowane po swojej stronie udziały. Brak chociażby jednego udziału spowoduje błąd interpolacji. W rezultacie zdalny serwer, aby wykonać poprawnie protokół, musi trwale przechowywać pełne dane użytkownika.

W pracy [B5] zaproponowano probabilistyczny schemat weryfikacji trwałości danych przechowywanych w chmurze obliczeniowej. Schemat ten nie wykorzystuje w swojej konstrukcji typowych kryptograficznych założeń związanych z trudnością obliczania dyskretnego logarytmu w grupach cyklicznych, ani założeń o trudności faktoryzacji dużych liczb. Schemat jest oparty jedynie na bezpiecznych, efektywnie obliczalnych, jednokierunkowych funkcjach skrótu modelowanych jako wyrocznie losowe. Użytkownik w fazie przygotowywania danych do wysłania na zdalną platformę generuje mały zbiór losowych wyzwań $\{c_i\}$. Dla każdego pliku $f$ przechowywanego w chmurze obliczany jest zbiór odpowiedzi $\{r_i = \mathcal{H}(f, c_i)\}$. Pary $\{c_i, r_i\}$ stanowią metadane pliku $f$ przesyłanego do chmury. Podstawowy krok weryfikacyjny polega na wysłaniu do chmury niewykorzystanego wyzwania $c_i$ i porównania otrzymanej odpowiedzi $\mathcal{H}(f, c_i)$, wyliczonej zdalnie, z wartością $r_i$ przechowywaną lokalnie. W pracy konstruktywnie wykazano istnienie efektywnych strategii odpytywania chmur obliczeniowych, o dane przechowywane zdalnie, za pomocą wyzwań $c_i$ będących częścią rzeczywistych metadanych, oraz wyzwań generowanych losowo, w taki sposób, aby z dużym prawdopodobieństwem wykrywać zmiany tych danych. Zaproponowany schemat wykorzystania tych strategii w oparciu o funkcje haszujące jest istotny przy założeniu możliwości łamania schematów opartych o DLP i RSA przez komputery kwantowe w bliskim horyzoncie czasowym. Schematy wykorzystujące jedynie funkcje skrótu są w tym kontekście bezpieczne.

W pracy [B6] analizowano wcześniejszą konstrukcję schematu z [46], służącą do zdalnego odpytywania bazy danych (przechowywanych w chmurze obliczeniowej), za pomocą pamięci podręcznej niewielkich rozmiarów, zaimplementowanej w bezpiecznym urządzeniu będącym interfejsem zdalnej platformy. Schemat oryginalny gwarantował, że wzorce odpytań oraz zaszyfrowanych odpowiedzi pozostają tajne i nie pozwalają adwersarzowi przeprowadzać skutecznego wnioskowania statystycznego. W pracy przeanalizowano protokół oryginalny i wykazano jego bezpieczeństwo w silniejszym modelu - z adwersarzem obserwującym wszystkie operacje zachodzące na zdalnym serwerze. Ponadto wskazano na potencjalne zagrożenia dotyczące prywatności zapytań i odpowiedzi, przy założeniu pewnych szczegółów implemetacyjnych przyjmowanych w pracy oryginalnej.

## 5.3 Inne schematy

### 5.3.1 Grupowe uwierzytelnianie urządzeń RFID

W pracy [B7] przeprowadzono analizę protokołu z [47] do równoległego uwierzytelniania grupy urządzeń małej mocy wzbudzanych radiowo (RFID). Przedyskutowano teoretyczne możliwości minimalizowania komunikatów wymienianych pomiędzy czytnikiem, a serwerem uwierzytelniania z bazą danych. W pracy wskazałem na nieprecyzyjne określenie zbioru urządzeń RFID uwierzytelnianych w komunikatach protokołu oryginalnego. Ponadto zaproponowałem ogólny schemat uwierzytelniania grupowego opierający się na złożeniu bezpiecznych protokłów uwierzytelniania dla pojedynczych urządzeń. Zaproponowałem model bezpieczeństwa dla takiej konstrukcji, oraz schemat specyficzny wykorzystujący filtry Blooma. W pracy udowodniono bezpieczeństwo tego schematu w przedstawionym modelu, oraz przeprowadzono analizę złożoności obliczeniowej i komunikacyjnej.

### 5.3.2 Schemat anonimowego uwierzytelniania oparty na funkcjach fizycznie nieklonowalnych

W pracy [B8] zaproponowałem anonimowy schemat identyfikacji oparty na funkcjach fizycznych nieklonowanych, będący rozszerzeniem rezultatu pracy [A4]. Użytkownik schematu przypisany jest do pewnej grupy i identyfikuje się jako jej członek przed weryfikatorem bez zdradzania swojej tożsamości. Bezpieczeństwo schematu i anonimowość użytkownika oparte są na założeniu nieklonowalności fizycznie realizowanej funkcji PUF. Schemat ten wykorzystuje infrastrukturę sprzętową podobną do zaproponowanej w [A4]. Użytkownik $u$ posiada urządzenie z modułem PUF, oznaczonym przez $P_u$, wraz z modułem pamięci na dane pomocnicze. Weryfikator podczas inicjalizacji grupy $L$ tworzy $T$ wielomianów $L_i$ stopnia $z$, oraz po dwa parametry $t, t'$ dla każdego z wielomianów. Zbiór $\{L_i(x), t_i, t'_i | i = 1, \ldots, T\}$ stanowi sekret weryfikatora dla grupy $L$, a każdy wielomian $L_i$ wykorzystywany jest tylko jeden raz w sesji $i$. Podczas rejestracji użytkownika w grupie $u$, dla każdego indeksu $i$ obliczane są kolejno wartości $x_{u,i} = P_u(t_i)$, $y_{u,i} = L_i(x_{u,i})$. Następnie wyliczana jest maska $y'_{u,i} = y_{u,i} + P_u(t'_i)$ zapisywana w module pamięci karty użytkownika. W wyniku tej procedury weryfikator posiada wektor $\langle x_{u,1}, \ldots, x_{u,T} \rangle$ dla użytkownika $u$. Po stronie użytkownika zapisany jest wektor $\left\langle y'_{u,1}, \ldots, y'_{u,T} \right\rangle$. Podczas identyfikacji użytkownika w sesji $i$ weryfikator definiuje niepełny zbiór interpolacyjny $\Psi = \{(x, L_i(x))\}$, dla losowego argumentu $x$, taki że brakuje w nim tylko jednego punktu, umożliwiającego interpolację wielomianu $L_i$. Ostatecznie tworzy wyzwanie $\langle i, t_i, t'_i, \Psi \rangle$ wysyłane do użytkownika. Użytkownik oblicza wartości $x_{u,i} = P_u(t_i)$, $y_{u,i} = y'_{u,i} - P_u(t'_i)$. Następnie tworzy pełny zbiór interpolacyjny $\Psi' = \Psi \cup \{(x_{u,i}, y_{u,i})\}$, za pomocą którego rekonstruowany jest wielomian $L_i$. Użytkownik wybiera losowo argument $x_r$ niewykorzystywany w $\Psi$ i oblicza $y_r = L_i(x_r)$. Para $x_r, y_r$ zwracana jest weryfikatorowi, który akceptuje użytkownika w grupie, jeśli $y_r = L_i(x_r)$ dla $L_i$ zapamiętanego po stronie weryfikatora.

### 5.3.3 Schemat głosowań elektronicznych

W pracy [B9] zaproponowano schemat głosowań elektronicznych gwarantujący wyborcy weryfikowalność oraz prywatność oddawanych głosów. Schemat ten, w sensie zapewniania anonimowości głosującym, jest podobny do schematów z prac [A2] oraz [B3], i opiera się na interpolacji Lagrange'a. Z każdą opcją wyboru związany jest pewien wielomian. Użytkownicy oddając głos na tą opcję wstawiają do jej zbioru interpolacyjnego swoje udziały. Stopień powstałego w ten sposób wielomianu odzwierciedla liczbę oddanych na daną opcje głosów. Weryfikacja uwzględnienia wyboru pojedynczego użytkownika polega na sprawdzeniu czy jego udział należy do opublikowanego wielomianu. Z kolei anonimowość wynika z faktu niemożności określenia za pomocą jakiego zbioru interpolacyjnego dany wielomian został zdefiniowany.

### 5.3.4 Znaczniki czasowe

W pracy [B10] zaproponowano schemat wystawiania znaczników czasowych, konstruowany jako łańcuch zmodyfikowanych podpisów Schnorra dla zadeklarowanych wcześniej parametrów. Dodanie nowego ogniwa w tym łańcuchu wiąże się z wystawieniem znacznika dla pewnego dokumentu oraz deklaracją nowych parametrów do wykorzystania w kolejnych ogniwach. Bezpieczeństwo schematu, gwarantowanego przez urząd certyfikujący, polega na niemożności wystawienia dwóch różnych znaczników dla tych samych - zadeklarowanych wcześniej parametrów. Każdy atak polegający na kolizji dwóch znaczników prowadzi do układu równań, których rozwiązaniem jest tajny klucz urzędu certyfikującego - który z założenia powinien pozostać sekretny.

### 5.3.5 Algorytmy dla systemów wykorzystujących krzywe eliptyczne

W pracy [B11] analizowano niedeterministyczne algorytmy mapowania dowolnych ciągów bitowych w punkty krzywych eliptycznych wykorzystywanych w kryptosystemach uwierzytelnionego ustalania kluczy sesyjnych. Praca ta jest rozszerzeniem pracy [48] dopuszczającym algorytmy probabilistyczne w funkcjach skrótu (propozycja oryginalna z [48] wykorzystywała jedynie algorytmy deterministyczne, o stałym czasie działania). Zaproponowane techniki pozwoliły na wyeliminowanie pewnych ataków czasowych na protokoły [49, 1]. Praca ta stanowi dopełnienie pracy [A10], oraz prac [A11, A12], związanych z analizą bezpieczeństwa protokołów dla bezprzewodowych dokumentów tożsamości.

### 5.3.6 Bezpieczna architektura sprzętowa - modele i protokoły

W pracy [B12] analizowano modele bezpieczeństwa dla schematów kryptograficznych implementowanych na urządzeniach użytkowników, z uwzględnieniem ataków przeprowadzanych przez adwersarzy posiadających wiedzę na temat architektury sprzętowej tych urządzeń. Zaproponowano podejście, w którym projektowane schematy wykorzystują modułową architekturę bezpieczeństwa, z wewnętrznym obszarami o różnym poziomie zabezpieczeń - dla różnych operacji arytmetycznych wykorzystujących różne klucze. Niezaniedbywalny sukces adwersarza określany w proponowanym modelu byłby związany ze zdarzeniem jednoczesnego przełamania wszystkich poziomów bezpieczeństwa we wszystkich obszarach, od najbardziej jawnych, tzw. zewnętrznych, do tych najbardziej chronionych - wewnętrznych, przy czym atak na obszary wewnętrzne byłby możliwy dopiero po przełamaniu obszarów zewnętrznych.

W pracy [B13] zaprezentowano koncepcję infrastruktury klucza publicznego opartą o tzw. lekkie certyfikaty kontrolowane przez użytkownika. Protokołowa warstwa realizowana jest za pomocą podpisów Schnorra z mediatorem, z addytywnym rozbiciem klucza tajnego. Dodatkowe mechanizmy bezpieczeństwa przy składaniu podpisu polegają na kontrolowanym generowaniu liczb pseudolosowych w wydzielonym obszarze pamięci urządzenia użytkownika, oraz na wykorzystaniu zobowiązań do losowych wartości efemerycznych. Ataki polegające na kolizji dwóch podpisów z tymi samymi wartościami prowadzą do kompromitacji długoterminowego klucza tajnego - wbrew gwarancjom producenta i administratora systemu (technika podobna do tej wykorzystanej w pracy [B10]).

W pracy [B14] przedstawiono metody kontroli procesu tworzenia podpisu w systemach wykorzystujących certyfikaty kluczy kryptograficznych dla urządzeń użytkownika. Problem z dotychczasowymi rozwiązaniami stosowanymi w praktyce jest związany z dość nierealistycznym założeniem, że zarówno urządzenie - od strony wykonywanej funkcjonalności (realizowanej sprzętowo i programowo), jak i sekretne klucze na nim składowane, podlegają pełnej kontroli użytkownika końcowego. Użytkownik zaś ufa w tym względzie producentowi i zaufanemu urzędowi certyfikującemu. W pracy przedyskutowano metody modyfikacji istniejących i uznanych schematów kryptograficznych, opartych o na trudności wyliczania logarytmu dyskretnego, które, wraz z odpowiednimi procedurami administracyjnymi, pozwoliłby na wdrożenie rozwiązań umożliwiających użytkownikowi większą kontrolę procesu tworzenia podpisów. W tym kontekście omówiono schematy podpisu z mediatorem, z synchronizowanymi zmianami tajnego wykładnika (klucza sekretnego), podpisy typu "fail-stop" [50], oraz techniki polegające na zobowiązaniach do losowych parametrów - bliskie technikom omawianym w pracach [B10, B13].

### 5.3.7 Sieci P2P

W pracy [B15] zajmowano się zagadnieniem równomiernego obciążenia węzłów w sieciach typu "peer-to-peer" (P2P), który modelowany jest jako problem generowania skończonych losowych podzbiorów odcinka $[0,1]$. Punkt podziału odcinka reprezentuje serwer sieci, a długość pododcinka zaczepionego w tym punkcie interpretowana jest jako wielkość obszaru adresowego danych składowanych na tym serwerze. Przeanalizowano dwie metody. Pierwsza, związana z protokołem Chord [51], wykorzystuje $n$ zmiennych losowych generowanych z rozkładu jednostajnego na odcinku $[0,1]$ do podziału tego odcinka na pododcinki. Druga, związana z protokołem CAN [52], wykorzystuje te zmienne do wskazania przedziału, który następnie dzielony jest na dwie równe części. W pracy pokazano, że wariancja długości odcinka w pierwszym przypadku wynosi w przybliżeniu $1/n^2$, zaś w drugim przypadku $(1/n^2)(1/\ln 2 - 1)$. Rezultat ten pozwala wnioskować, że protokół CAN bardziej równomiernie obciąża węzły danymi, niż porównywany w pracy protokół Chord.

## Bibliografia A (prace stanowiące podstawę rozprawy)

[A1] Cichoń, J., Krzywiecki, Ł., Kutyłowski, M., Wlaź, P.: Anonymous Distribution of Encryption Keys in Cellular Broadcast Systems. In: MADNES. Volume 4074 of Lecture Notes in Computer Science., Springer (2005) 96–109

[A2] Krzywiecki, Ł., Kutyłowski, M., Nikodem, M.: General Anonymous Key Broadcasting via Lagrangian Interpolation. IET Information Security **2**(3) (2008) 79–84

[A3] Krzywiecki, Ł., Kubiak, P., Kutyłowski, M.: A Revocation Scheme Preserving Privacy. In: Inscrypt. Volume 4318 of Lecture Notes in Computer Science., Springer (2006) 130–143

[A4] Krzywiecki, Ł., Kutyłowski, M.: Coalition Resistant Anonymous Broadcast Encryption Scheme Based on PUF. In: TRUST. Volume 6740 of Lecture Notes in Computer Science., Springer (2011) 48–62

[A5] Krzywiecki, Ł.: Schnorr-Like Identification Scheme Resistant to Malicious Subliminal Setting of Ephemeral Secret. In: SECITC. Volume 10006 of Lecture Notes in Computer Science. (2016) 137–148

[A6] Łukasz Krzywiecki and Mirosław Kutyłowski: Security of Okamoto Identification Scheme: a Defense against Ephemeral Key Leakage and Setup. In: SCC@AsiaCCS, ACM (2017) 43–50

[A7] Krzywiecki, Ł., Wszoła, M., Kutyłowski, M.: Brief Announcement: Anonymous Credentials Secure to Ephemeral Leakage. In: CSCML. Volume 10332 of Lecture Notes in Computer Science., Springer (2017) 96–98

[A8] Krzywiecki, Ł., Słowik, M.: Strongly Deniable Identification Schemes Immune to Prover's and Verifier's Ephemeral Leakage. In: SECITC. Volume 10543 of Lecture Notes in Computer Science., Springer (2017) 115–128

[A9] Kutyłowski, M., Krzywiecki, Ł., Kubiak, P., Koza, M.: Restricted Identification Scheme and Diffie-Hellman Linking Problem. In: INTRUST. Volume 7222 of Lecture Notes in Computer Science., Springer (2011) 221–238

[A10] Hanzlik, L., Krzywiecki, Ł., Kutyłowski, M.: Simplified PACE|AA Protocol. In: ISPEC. Volume 7863 of Lecture Notes in Computer Science., Springer (2013) 218–232

[A11] Hanzlik, L., Kluczniak, K., Krzywiecki, Ł., Kutyłowski, M.: Mutual Chip Authentication. In: TrustCom/ISPA/IUCC, IEEE Computer Society (2013) 1683–1689

[A12] Hanzlik, L., Kluczniak, K., Kutyłowski, M., Krzywiecki, Ł.: Mutual Restricted Identification. In: EuroPKI. Volume 8341 of Lecture Notes in Computer Science., Springer (2013) 119–133

[A13] Krzywiecki, Ł., Kluczniak, K., Kozieł, P., Panwar, N.: Privacy-oriented dependency via deniable SIGMA protocol. Computers & Security **79** (2018) 53–67

[A14] Łukasz Krzywiecki and Tomasz Wlislocki: Deniable Key Establishment Resistance Against eKCI Attacks. Security and Communication Networks **2017** (2017) 7810352:1–7810352:13

[A15] Dolev, S., Krzywiecki, Ł., Panwar, N., Segal, M.: Vehicle Authentication via Monolithically Certified Public Key and Attributes. Wireless Networks **22**(3) (2016) 879–896

[A16] Dolev, S., Krzywiecki, Ł., Panwar, N., Segal, M.: Dynamic Attribute Based Vehicle Authentication. Wireless Networks **23**(4) (2017) 1045–1062

[A17] Dolev, S., Krzywiecki, Ł., Panwar, N., Segal, M.: Optical PUF for Non-Forwardable Vehicle Authentication. Computer Communications **93** (2016) 52–67

# Bibliografia B (prace stanowiące dodatkowy dorobek naukowy)

[B1] Klonowski, M., Krzywiecki, Ł., Kutyłowski, M., Lauks, A.: Step-Out Ring Signatures. In: MFCS. Volume 5162 of Lecture Notes in Computer Science., Springer (2008) 431–442

[B2] Krzywiecki, Ł., Sulkowska, M., Zagórski, F.: Hierarchical Ring Signatures Revisited - Unconditionally and Perfectly Anonymous Schnorr Version. In: SPACE. Volume 9354 of Lecture Notes in Computer Science., Springer (2015) 329–346

[B3] Klonowski, M., Krzywiecki, Ł., Kutyłowski, M., Lauks, A.: Step-out Group Signatures. Computing **85**(1-2) (2009) 137–151

[B4] Krzywiecki, Ł., Kutyłowski, M.: Proof of Possession for Cloud Storage via Lagrangian Interpolation Techniques. In: NSS. Volume 7645 of Lecture Notes in Computer Science., Springer (2012) 305–319

[B5] Krzywiecki, Ł., Majcher, K., Macyna, W.: Efficient Probabilistic Methods for Proof of Possession in Clouds. In: DMBD. Volume 9714 of Lecture Notes in Computer Science., Springer (2016) 364–372

[B6] Krzywiecki, Ł., Kutyłowski, M., Misztela, H., Struminski, T.: Private Information Retrieval with a Trusted Hardware Unit - Revisited. In: Inscrypt. Volume 6584 of Lecture Notes in Computer Science., Springer (2010) 373–386

[B7] Błaśkiewicz, P., Krzywiecki, Ł., Syga, P.: RFID Tags Batch Authentication Revisited - Communication Overhead and Server Computational Complexity Limits. In: ISPEC. Volume 10060 of Lecture Notes in Computer Science. (2016) 209–223

[B8] Krzywiecki, Ł.: Anonymous Authentication Scheme Based on PUF. In: ICISC. Volume 9558 of Lecture Notes in Computer Science., Springer (2015) 359–372

[B9] Krzywiecki, Ł., Kutyłowski, M.: Lagrangian E-Voting: Verifiability on Demand and Strong Privacy. In: TRUST. Volume 6101 of Lecture Notes in Computer Science., Springer (2010) 109–123

[B10] Krzywiecki, Ł., Kubiak, P., Kutyłowski, M.: Stamp and Extend - Instant But Undeniable Timestamping Based on Lazy Trees. In: INTRUST. Volume 7711 of Lecture Notes in Computer Science., Springer (2012) 5–24

[B11] Krzywiecki, Ł., Kubiak, P., Kutyłowski, M.: Probabilistic Admissible Encoding on Elliptic Curves - Towards PACE with Generalized Integrated Mapping. In: SOFSEM. Volume 8327 of Lecture Notes in Computer Science., Springer (2014) 395–406

[B12] Kutyłowski, M., Hanzlik, L., Kluczniak, K., Kubiak, P., Krzywiecki, Ł.: Forbidden City Model - Towards a Practice Relevant Framework for Designing Cryptographic Protocols. In: ISPEC. Volume 8434 of Lecture Notes in Computer Science., Springer (2014) 42–59

[B13] Krzywiecki, Ł., Kubiak, P., Kutyłowski, M., Tabor, M., Wachnik, D.: Lightweight Certificates - Towards a Practical Model for PKI. In: BIS. Volume 117 of Lecture Notes in Business Information Processing., Springer (2012) 296–307

[B14] Kutyłowski, M., B laśkiewicz, P., Krzywiecki, Ł., Kubiak, P., Paluszynski, W., Tabor, M.: Technical and Legal Meaning of Śole Control- Towards Verifiability in Signing Systems. In: BIS (Workshops). Volume 97 of Lecture Notes in Business Information Processing., Springer (2011) 270–281

[B15] Cichoń, J., Klonowski, M., Krzywiecki, Ł., Różański, B., Źieliński, P.: Random Subsets of the Interval and P2P Protocols. In: APPROX-RANDOM. Volume 4627 of Lecture Notes in Computer Science., Springer (2007) 409–421

# Literatura

[1] Bender, J., Dagdelen, Ö., Fischlin, M., Kügler, D.: The PACE|AA Protocol for Machine Readable Travel Documents, and Its Security. In: Financial Cryptography and Data Security: 16th International Conference, FC 2012, Kralendijk, Bonaire, Februray 27-March 2, 2012, Revised Selected Papers. Springer Berlin Heidelberg, Berlin, Heidelberg (2012) 344–358

[2] ISO/IEC JTC1 SC17 WG3/TF5 for the International Civil Aviation Organization: Supplemental access control for machine readable travel documents - Technical Report, (March 08, 2011)

[3] Bundesamt für Sicherheit in der Informationstechnik: Technical guideline: Tr-03110 advanced security mechanisms for machine readable travel documents. v. 2.05. 2010. (2010)

[4] Krzywiecki, Ł.: Deniable Version of SIGMA Key Exchange Protocol Resilient to Ephemeral Key Leakage. In: ProvSec. Volume 8782 of Lecture Notes in Computer Science., Springer (2014) 334–341

[5] Krawczyk, H.: SIGMA: The 'SIGn-and-MAc' Approach to Authenticated Diffie-Hellman and Its Use in the IKE-Protocols. In: CRYPTO. (2003) 400–425

[6] Krawczyk, H.: Hmqv: A high-performance secure diffie-hellman protocol. In Shoup, V., ed.: CRYPTO. Volume 3621 of Lecture Notes in Computer Science., Springer (2005) 546–566

[7] Tang, Q., Chen, L.: Extended kci attack against two-party key establishment protocols. Inf. Process. Lett. **111**(15) (2011) 744–747

[8] Shlomi Dolev, Nisha Panwar, Michael Segal, Łukasz Krzywiecki: Certificating Vehicle Public Key with Vehicle Attributes. US Patent 9769658

[9] Fiat, A., Naor, M.: Broadcast Encryption. In Stinson, D.R., ed.: CRYPTO. Volume 773 of LNCS., Springer (1993) 480–491

[10] Naor, M., Pinkas, B.: Efficient Trace and Revoke Schemes. In Frankel, Y., ed.: Financial Cryptography. Volume 1962 of LNCS., Springer (2000) 1–20

[11] Matsuzaki, N., Anzai, J., Matsumoto, T.: Light Weight Broadcast Exclusion Using Secret Sharing. In Dawson, E., Clark, A., Boyd, C., eds.: ACISP. Volume 1841 of LNCS., Springer (2000) 313–327

[12] Yoo, E.S., Jho, N.S., Cheon, J.H., Kim, M.H.: Efficient Broadcast Encryption Using Multiple Interpolation Methods. In Park, C., Chee, S., eds.: ICISC. Volume 3506 of LNCS., Springer (2004) 87–103

[13] Tzeng, W.G., Tzeng, Z.J.: A Public-Key Traitor Tracing Scheme with Revocation Using Dynamic Shares. In Kim, K., ed.: Public Key Cryptography. Volume 1992 of LNCS., Springer (2001) 207–224

[14] Dodis, Y., Fazio, N., Kiayias, A., Yung, M.: Scalable public-key tracing and revoking. In Rajsbaum, S., Herzberg, A., eds.: PODC '03: Proceedings of the 22nd Annual Symposium on Principles of Distributed Computing, ACM Press (2003) 190–199

[15] Dodis, Y., Fazio, N.: Public Key Trace and Revoke Scheme Secure against Adaptive Chosen Ciphertext Attack. In Desmedt, Y., ed.: Public Key Cryptography. Volume 2567 of LNCS., Springer (2003) 100–115

[16] Kim, C.H., Hwang, Y.H., Lee, P.J.: Practical Pay-TV Scheme Using Traitor Tracing Scheme for Multiple Channels. In Lim, C.H., Yung, M., eds.: WISA. Volume 3325 of LNCS., Springer (2004) 264–277

[17] Watanabe, Y., Numao, M.: Multi-round Secure-Light Broadcast Exclusion Protocol with Pre-processing. In Snekkenes, E., Gollmann, D., eds.: ESORICS. Volume 2808 of LNCS., Springer (2003) 85–99

[18] Cole, R., Frieze, A., Maggs, B.M., Mitzenmacher, M., Richa, A.W., Sitaraman, R., Upfal, E. In: On Balls and Bins with Deletions. Springer Berlin Heidelberg, Berlin, Heidelberg (1998) 145–158

[19] Azar, Y., Broder, A.Z., Karlin, A.R., Upfal, E.: Balanced allocations. SIAM Journal on Computing **29**(1) (1999) 180–200

[20] Mitzenmacher, M., Prabhakar, B., Shah, D.: Load balancing with memory. In: In Proc. of the 43rd IEEE Symp. on Foundations of Computer Science (FOCS. (2002)

[21] Berenbrink, P., Czumaj, A., Steger, A., Vöcking, B.: Balanced allocations: The heavily loaded case. SIAM Journal on Computing **35**(6) (2006) 1350–1385

[22] Vöcking, B.: How asymmetry helps load balancing. J. ACM **50**(4) (2003) 568–589

[23] Hwang, Y.H., Kim, C.H., Lee, P.J.: An Efficient Revocation Scheme with Minimal Message Length for Stateless Receivers. In Safavi-Naini, R., Seberry, J., eds.: ACISP. Volume 2727 of LNCS., Springer (2003) 377–386

[24] Watanabe, Yuji and Numao, Masayuki. In: Multi-round Secure Light-Weight Broadcast Exclusion Protocol with Pre-processing. Springer Berlin Heidelberg, Berlin, Heidelberg (2003) 85–99

[25] Barth, A., Boneh, D., Waters, B.: Privacy in encrypted content distribution using private broadcast encryption. In: Financial Cryptography and Data Security: Tenth International Conference, February'06, Proceedings. LNCS, Springer (2006)

[26] Fiat, A., Shamir, A.: How To Prove Yourself: Practical Solutions to Identification and Signature Problems. In: Advances in Cryptology — CRYPTO' 86: Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg (1987) 186–194

[27] Feige, U., Fiat, A., Shamir, A.: Zero-knowledge proofs of identity. Journal of Cryptology **1**(2) 77–94

[28] Guillou, L.C., Quisquater, J.J.: A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory. In: Lecture Notes in Computer Science on Advances in Cryptology-EUROCRYPT'88, New York, NY, USA, Springer-Verlag New York, Inc. (1988) 123–128

[29] Schnorr, C.P.: Efficient signature generation by smart cards. J. Cryptology **4**(3) (1991) 161–174

[30] Okamoto, T.: Provably Secure and Practical Identification Schemes and Corresponding Signature Schemes. In: Advances in Cryptology — CRYPTO' 92: 12th Annual International Cryptology Conference Santa Barbara, California, USA August 16–20, 1992 Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg (1993) 31–53

[31] Kurosawa, K., Heng, S.H.: Identity-Based Identification Without Random Oracles. In: Computational Science and Its Applications – ICCSA 2005: International Conference, Singapore, May 9-12, 2005, Proceedings, Part II. Springer Berlin Heidelberg, Berlin, Heidelberg (2005) 603–613

[32] Kurosawa, K., Heng, S.H.: The Power of Identification Schemes. In: Public Key Cryptography - PKC 2006: 9th International Conference on Theory and Practice in Public-Key Cryptography, New York, NY, USA, April 24-26, 2006. Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg (2006) 364–377

[33] Alwen, J., Dodis, Y., Wichs, D.: Leakage-Resilient Public-Key Cryptography in the Bounded-Retrieval Model. In: Advances in Cryptology - CRYPTO 2009: 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg (2009) 36–54

[34] Canetti, R., Goldreich, O., Goldwasser, S., Micali, S.: Resettable zero-knowledge (extended abstract). In: Proceedings of the Thirty-second Annual ACM Symposium on Theory of Computing. STOC '00, New York, NY, USA, ACM (2000) 235–244

[35] Bellare, M., Fischlin, M., Goldwasser, S., Micali, S.: Identification Protocols Secure against Reset Attacks. In: Advances in Cryptology — EUROCRYPT 2001: International Conference on the Theory and Application of Cryptographic Techniques Innsbruck, Austria, May 6–10, 2001 Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg (2001) 495–511

[36] Camenisch, J., Lysyanskaya, A.: Signature schemes and anonymous credentials from bilinear maps. In: Annual International Cryptology Conference, Springer (2004) 56–72

[37] Lysyanskaya, A., Rivest, R.L., Sahai, A., Wolf, S.: Pseudonym systems. In Heys, H.M., Adams, C.M., eds.: Selected Areas in Cryptography, 6th Annual International Workshop, SAC'99, Kingston, Ontario, Canada, August 9-10, 1999, Proceedings. Volume 1758 of Lecture Notes in Computer Science., Springer (1999) 184–199

[38] Krawczyk, H.: SIGMA: The 'SIGn-and-MAc' Approach to Authenticated Diffie-Hellman and Its Use in the IKE Protocols. In: Advances in Cryptology - CRYPTO 2003: 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003. Proceedings. Springer Berlin Heidelberg, Berlin, Heidelberg (2003) 400–425

[39] Canetti, R., Krawczyk, H.: Analysis of key-exchange protocols and their use for building secure channels. In Pfitzmann, B., ed.: EUROCRYPT. Volume 2045 of Lecture Notes in Computer Science., Springer (2001) 453–474

[40] Raimondo, M.D., Gennaro, R., Krawczyk, H.: Deniable authentication and key exchange. In Juels, A., Wright, R.N., di Vimercati, S.D.C., eds.: ACM Conference on Computer and Communications Security, ACM (2006) 400–409

[41] Krawczyk, H.: SKEME: a versatile secure key exchange mechanism for Internet. In Ellis, J.T., Neuman, B.C., Balenson, D.M., eds.: NDSS, IEEE Computer Society (1996) 114–127

[42] Herranz, J., Sáez, G.: Forking lemmas for ring signature schemes. In Johansson, T., Maitra, S., eds.: INDOCRYPT. Volume 2904 of Lecture Notes in Computer Science., Springer (2003) 266–279

[43] Boneh, D., Lynn, B., Shacham, H.: Short signatures from the weil pairing. J. Cryptology **17**(4) (2004) 297–319

[44] ISO/IEC IS 9798-3: Entity authentication mechanisms, Part 3: Entity authentication using asymmetric techniques (1993)

[45] Ustaoglu, B.: Obtaining a secure and efficient key agreement protocol from (h)mqv and naxos. Des. Codes Cryptography **46**(3) (2008) 329–342

[46] Yang, Y., Ding, X., Deng, R.H., Bao, F.: An efficient PIR construction using trusted hardware. In Wu, T., Lei, C., Rijmen, V., Lee, D., eds.: Information Security, 11th International Conference, ISC 2008, Taipei, Taiwan, September 15-18, 2008. Proceedings. Volume 5222 of Lecture Notes in Computer Science., Springer (2008) 64–79

[47] Chen, J., Miyaji, A., Su, C.: A Provable Secure Batch Authentication Scheme for EPCGen2 Tags. In: Provable Security: 8th Int. Conference, ProvSec 2014. Springer International Publishing (2014) 103–116

[48] Brier, E., Coron, J.S., Icart, T., Madore, D., Randriam, H., Tibouchi, M.: Efficient Indifferentiable Hashing into Ordinary Elliptic Curves. In Rabin, T., ed.: Advances in Cryptology – CRYPTO 2010, Berlin, Heidelberg, Springer Berlin Heidelberg (2010) 237–254

[49] Bender, J., Dagdelen, Ö., Fischlin, M., Kügler, D.: The pace|aa protocol for machine readable travel documents, and its security. In Keromytis, A.D., ed.: Financial Cryptography and Data Security - 16th International Conference, FC 2012, Kralendijk, Bonaire, Februray 27-March 2, 2012, Revised Selected Papers. Volume 7397 of Lecture Notes in Computer Science., Springer (2012) 344–358

[50] Bleumer, G.: Fail-stop signature. In van Tilborg, H.C.A., Jajodia, S., eds.: Encyclopedia of Cryptography and Security, 2nd Ed. Springer (2011) 446–447

[51] Stoica, I., Morris, R.T., Karger, D.R., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable peer-to-peer lookup service for internet applications. In: SIGCOMM. (2001) 149–160

[52] Ratnasamy, S., Francis, P., Handley, M., Karp, R., Shenker, S.: A scalable content-addressable network. SIGCOMM Comput. Commun. Rev. **31**(4) (August 2001) 161–172