Algebraic Cryptography: Ideas, Proposals, Vulnerabilities and Possibilities

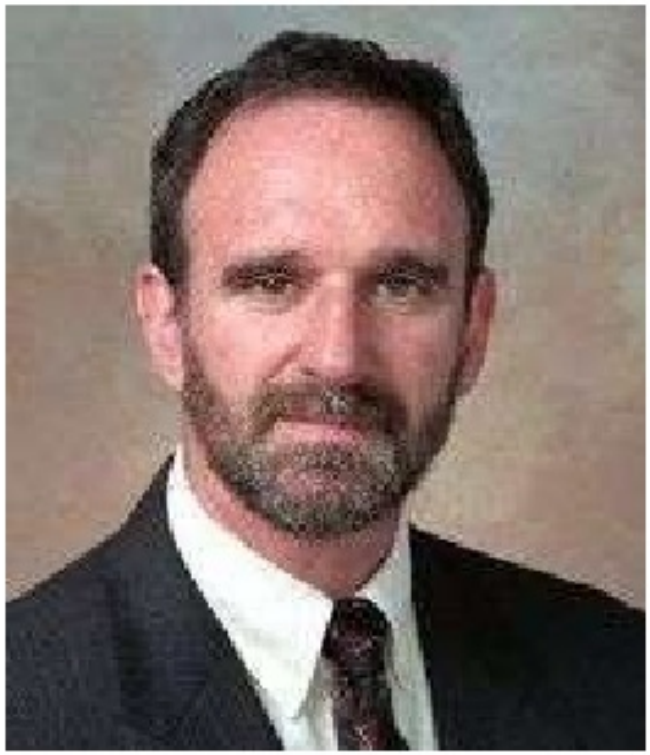
Vitaly Roman'kov

OMSK: ALMAZ, April 2015

Content

Pioneers of Public – Key Cryptography

Whitfield Diffie



Martin Hellman

Merkle

Ralph Merkle

Diffie-Hellman-Merkle scheme

The Diffie-Hellman-Merkle (1976) key agreement scheme

- Public data: $\{G$ – group, $g \in G\}$.
- Alice chooses a private $k \in \mathbb{Z}$, then publics g^k.
- Bob chooses a private $l \in \mathbb{Z}$, then publics g^l.
- Agreement:

$$Alice : (g^l)^k = g^{kl} = (g^k)^l : Bob$$

ElGamal

The ElGamal (1980)–Massey-Omura (1982) cryptosystem for message (key) transmission

- Private data (key): $\{g \in G\}$. Public data: group G and a positive integer r such that $g^r = 1$. Number r may be given as the order $|G|$ of G, or $|g|$ of g.
- Alice chooses a private $k \in \mathbb{Z}, (k, r) = 1$, then publics g^k.
- Bob chooses $l \in \mathbb{Z}, (l, r) = 1$, then computes and publics $(g^k)^l = g^{kl}$.
- Alice computes $k^{-1} (\mathrm{mod} r)$ and then publics $(g^{kl})^{k^{-1}} = g^l$.
- The transmitted key: Bob computes $l^{-1} (\mathrm{mod} r)$, then he recovers the transmitted key:

$$(g^l)^{l^{-1}} = g.$$

ElGamal (1985) cryptosystem for message (key) transmission

- Alice sets public data: $\{G$ – group, $g \in G\}$. Also she sets private data (key): $0 < a < |g|$, and other public data (encypering key): g^a.
- Bob wants to send a message $m \in G$ to Alice. He chooses a private $k \in \mathbb{Z}, 0 < k < |g|$, then publics (g^k, mg^{ak}).
- The transmitted message: Alice computes $g^{ak} = (g^k)^a$ and $(g^{ak})^{-1}$, then recovers the message

$$m = (mg^{ak})(g^{ak})^{-1}.$$

Platforms and operations: number theoretic and algebraic

- Classic platforms: $G = \mathbb{F}_{p^r}^*$ – the multiplicative group of a finite field $\mathbb{F}_{p^r}$, or $G(E)$ – the group of an elliptic curve E (over a finite field).
- Classic operations: multiplication and involution, or addition and taking multiple.
- Group based cryptography platforms: G – abstract group (Artin braid groups, matrix groups over fields and rings, polycyclic groups, finite p-groups are most popular).
- Group based operations: Right (left) multiplication, inversion, involution, conjugation, actions by endomorphism (automorphism).

Algebraic operations: operations derived from basic operations of the given algebraic platform, including morphisms.

The Discrete Logarithm Problem in a matrix group

The Discrete Logarithm Problem in $GL_n(\mathbb{F}_q), q = p^r$.
$g \in GL_n(\mathbb{F}_q), h = g^k$, $k = log_g h$.

- Find the Jordan form: $J(g) = tgt^{-1}$.
- $J(g) = J_{r_1}(\lambda_1) \oplus ... \oplus J_{r_t}(\lambda_t)$, $\sum_{i=1}^{t} r_i = n$.
- $\lambda_1, ..., \lambda_t$ are roots (in extensions $\mathbb{F}_{q^{n_i}}$ of $\mathbb{F}_q$) of the characteristic polynomial
  $p_g(x) = |g - \lambda E| = (x - \lambda_1)^{r_1} ... (x - \lambda_t)^{r_t} = 0$.

Here

$$J_s(\lambda) = \begin{pmatrix} \lambda & 1 & 0 & ... & 0 \\ 0 & \lambda & 1 & ... & 0 \\ . & . & . & . & . \\ 0 & 0 & ... & 0 & \lambda \end{pmatrix}$$

is Jordan block of size s.

Effective computation of the Jordan form

Effective computation of $J(g)$:
Input: $g \in GL_n(\mathbb{F}_q)$.
Output: $J(g)$.

1. By the Hessenberg's algorithm (which is more effective in the case of a finite field than the deterministic algorithm which is $O(n^3)$) we find the characteristic polynomial $p_g(x)$.
2. By the probabilistic polynomial Ben-Or's algorithm we get a presentation $p_g(x) = f_1^{e_1} \ldots f_s^{e_s}$, where f_i is irreducible polynomial over $\mathbb{F}_q$ of degree n_i.
3. $\mathbb{F}_{q^{n_i}} = \mathbb{F}_q[x]/ideal(f_i(x))$. We find roots $\alpha_{ij}, 1 \leq j \leq n_i$, of f_i in $\mathbb{F}_{q^{n_i}}$. Namely, $\alpha_{i1} = x, \alpha_{ij} = x^{q^{j-1}} mod(f_i(x)), 2 \leq j \leq n_i$.
4. We find sizes of Jordan blocks J_l, and then we get the Jordan form $J(g) = J_1 \oplus \ldots \oplus J_t$.

Polynomiality of the proposed algorithm

We know that the Hessenberg's and Ben-Or's algorithms solve tasks in polynomial time. Because $n_i \leq n$, we use in each of $s \leq n$ iterations on the step 3 $log\ q^{n_i} \leq nlog\ q$ operations. Hence the time complexity of this procedure is estimated by a polynomial in n and $log\ q$.

Reduction of the discrete logarithm problem for a matrix group over a finite field to the multiple discrete logarithm problem for finite field(s)

Reduction of the DLP for a matrix group to the multiple DLP for finite fields:
Input: $h = g^k, h, g \in GL_n(\mathbb{F}_q)$.
Output: $k \in \mathbb{Z}$.

1. Find t such that $tgt^{-1} = J(g)$.
2. Compute $tht^{-1} = (tgt^{-1})^k$.
3. $\alpha_{ij}^k = \beta_{ij}$, where β_{ij} are corresponding diagonal entries of tht^{-1}.

Idea: A.J. Menezes and S.A. Vanstone, A note on cyclic groups, finite fields, and the discrete logarithm problem, Appl. Algebra in Engineering, Communication and Computing, 1992, 3, 67-74. A.J. Menezes and Y.-H. Wu, The discrete logarithm problem in GL(n, q). Ars Combinatoria, 1997, 47, 23-32.

Extra equations in the case of Jordan blocks of size ≥ 2.

Example

$$\begin{pmatrix} \alpha & 1 \\ 0 & \alpha \end{pmatrix}^k = \begin{pmatrix} \alpha^k & k\alpha^{k-1} \\ 0 & \alpha^k \end{pmatrix}.$$

Extra equation: $k\alpha^{k-1} = \beta$, where β is 12-entry of $t^{-1}ht$. By multiplying of its both sides to α, and dividing to α^k we obtain $k \in \mathbb{F}_q$, namely, $k \bmod p$, i.e., we get extra useful information.

Conclusion

The logarithm problem in $\mathrm{GL}_n(\mathbb{F}_q)$ is no more difficult than the logarithm problem in a suitable extension $\mathbb{F}_{q^m}$ where $m \leq n$.

Other generalizations of the Diffie-Hellman-Merkle scheme: mixed (involution and conjugation) version

Involution and conjugation:
Public data: group G, element $g \in G$, and subgroups $H_1, H_2 \leq G$, such that $[H_1, H_2] = 1$.

- Alice chooses a private number $k \in \mathbb{Z}$ and element $a \in H_1$, and then publics $(g^k)^a$.
- Bob chooses a private number $l \in \mathbb{Z}$ and element $b \in H_2$, and then publics $((g^l)^b)$.
- Shared key is

$$Alice : (((g^l)^b)^k)^a = \quad (g^{kl})^{ab} \quad = (((g^k)^a)^l)^b : Bob$$

Preference of the mixed version

Because conjugation can permute Jordan blocks the Menezes-Vanstone-Wu's algorithm to compute k and l can not be applied.

Other generalizations of the Diffie-Hellman-Merkle's scheme: versions using automrphisms or endomorphisms

Action by an automorphism or endomorphism:

Public data: group G, element $g \in G$, and subgroups $H_1, H_2 \leq \mathrm{Aut}\, G$, such that $[H_1, H_2] = 1$.

- Alice chooses a private automorphism $\varphi \in H_1$, and then publics $\varphi(g)$.
- Bob chooses a private automorphism $\psi \in \mathrm{Aut}\, G$, and then publics $\psi(g)$.
- Shared key is

$$Alice : \varphi(\psi(g)) = \quad \psi(\varphi(g)) : Bob$$

Particular case: $\varphi, \psi \in \mathrm{Inn}\, G$. This is Diffie-Hellman-Merkle scheme.

One can use $\mathrm{End}\, G$ instead of $\mathrm{Aut}\, G$, and two commuting elementwise subsemigroups H_1, H_2 of $\mathrm{End}\, G$, as well.

Cheon and Jung versus Ko and al.

In [A polynomial time algorithm for the braid Diffie-Hellman conjugacy problem, Advances in Cryptology, CRYPTO'2003, Lect. Notes in Comp. Science, 7-14] Cheon and Jun proposed a polynomial probabilistic algorithm solving the Ko et. al. key exchange protocol via faithful representation of the Artin braid group B_n on n strings by matrices.

Conjugation instead of involution

Conjugation instead of involution:

Public data: group G and element $g \in G$, subgroups H_1 and H_2 such that $[H_1, H_2] = 1$.

- Alice chooses a private element $a \in H_1$, then publics $g^a = aga^{-1}$.
- Bob chooses a private element $b \in H_2$, then publics g^b.
- Shared key is:

$$Alice : (g^b)^a = g^{ab} = K = g^{ba} = (g^a)^b : Bob$$

The search conjugacy problem

G should be nonabelian. Security of the scheme bases on the conjugacy search problem, that is to find the element a by g and g^a, or, more generally, to find an element $a' \in H_1$ for which one has $g^a = g^{a'}$. Then one can compute
$(g^b)^{a'} = (g^{a'})^b = (g^a)^b = g^{ab} = K$.

Cryptanalysis by Cheon and Jun

How one can find a conjugating element:

1. Apply the Lawrence-Krammer representation $\varphi : B_n \to GL_{n(n-1)/2}(\mathbb{Z}[t^{\pm 1}, q^{\pm 1}])$ to compute images $\varphi(g), \varphi(a^{-1}ga)$.
2. Find $\varphi(a)$ such that $\varphi(a)\varphi(g)\varphi(a)^{-1} = \varphi(g^a)$.
3. Find the preimage $a \in B_n$ of $\varphi(a)$.

Difficulties

Difficulties:

1. Direct applications of Gauss elimination should deal with coefficients as large as 2^{2^n}).
2. A solution "$\varphi(a)$" can be out of $\varphi(H_1)$. Notice, that $\varphi(a)$ should be invertible.

Hence, this approach is irreal.

Advanced cryptanalysis by Cheon and Jun

How one can find the shared key:

1. Apply the Lawrence-Krammer representation $\varphi : B_n \to GL_{n(n-1)/2}(\mathbb{Z}[t^{\pm 1}, q^{\pm 1}])$ to compute images $\varphi(g), \varphi(g^a), varphi(g^b)$.
2. Solve equations $\varphi(g^a)Y = Y\varphi(g), \varphi(\sigma_i)Y = Y\varphi(\sigma_i)$.
3. Find a preimage a of $\varphi(a)$ in B_n.

Now one has similar difficulties in applying of the cryptanalysis. One has too much equations and unknowns. If one finds a singular solution, he needs to repeat procedure. Thus this algorithm is probabilistic. It is practically non realizable.

Tsaban

Tsaban versus all

In [Practical polynomial time solutions of several major problems in noncommutative-algebraic cryptography, Cryptology ePrint Archive: Report 2014/041] Boaz Tsaban provided provable polynomial time solutions of a number of problems in algebraic cryptography. He named this approach the algebraic span method. Now we demonstrate the method with applying it to the M. Anshel et al. key exchange protocol.

M. Anshel et al. key exchange protocol

M. Anshel et al. key exchange protocol:

- Public data: group G and elements $a_1, ..., a_k, b_1, ..., b_k \in G$.
- Alice chooses a group word $v(x_1, ..., x_k)$, computes $a = v(a_1, ..., a_k)$, and publics $b_1^a, ..., b_k^a$.
- Bob chooses a group word $w(x_1, ..., x_k)$, computes $b = w(b_1, ..., b_k)$, and publics $a_1^b, ..., a_k^b$.
- The shared key K is the commutator $[a, b] = aba^{-1}b^{-1}$. Alice can compute K as $av(a_1^b, ..., a_k^b)^{-1}$. Bob computes K as $w(b_1^a, ..., b_k^a)b^{-1}$.

Tsaban's cryptanalysis of M. Anshel et. al. protocol:

$G = \mathrm{gp}(g_1, ..., g_k)$ is assumed to be a finitely generated matrix group over a finite field $\mathbb{F}_q, q = p^r$.
For a set $S \subseteq \mathrm{M}_n(\mathbb{F}_q)$, let $\mathrm{Alg}(S)$ be the algebra generated by S.
Then $\mathrm{Alg}(G) = \mathrm{span}(G)$. A basis for the underlying vector space of $\mathrm{Alg}(G)$ can be computed in time $O(kn^6)$.

Tsaban's cryptanalysis of M. Anshel et. al. protocol

Input: $a_1, ..., a_k, b_1, ..., b_k, a_1^b, ..., a_k^b, b_1^a, ..., b_k^a \in G$, where $a \in \mathrm{gp}(a_1, ..., a_k), b \in \mathrm{gp}(b_1, ..., b_k)$ are unknown.

1. Offline: Generate bases for $\mathrm{Alg}(A)$ and $\mathrm{Alg}(B)$. Let d be the maximum of the sizes of these bases.
2. Online:

   (a) Solve the following homogeneous system of linear equations on the d coefficients determing $x \in \mathrm{Alg}(A)$:

   $$b_i \cdot x = x \cdot b_i^a \text{ for } i = 1, ..., k.$$

   (b) Fix a basis for the solution space, and pick random solutions x until x is invertible.

   (c) Solve the following homogeneous system of linear equations on the d coefficients determing $x \in \mathrm{Alg}(B)$:

   $$a_i \cdot y = y \cdot a_i^b \text{ for } i = 1, ..., k.$$

   (d) Fix a basis for the solution space, and pick random solutions y until y is invertible.

   (e) Output: $K = xyx^{-1}y^{-1}$.

The speaker and Myasnikov versus all

Основная идея метода линейного разложения
Обозначения:
V – пространство конечной размерности над полем $\mathbb{F}$ с базисом $\mathcal{B} = \{v_1, \ldots, v_r\}$.
$\mathrm{End}(V)$ – полугруппа эндоморфизмов пространства V.
Элементы $v \in V$– векторы относительно базиса $\mathcal{B}$.
Эндоморфизмы $a \in \mathrm{End}(V)$ – матрицы относительно $\mathcal{B}$, v^a – образ v относительно a.
Для подмножеств $W \subseteq V$ и $A \subseteq \mathrm{End}(V)$ обозначим $W^A = \{w^a | w \in W, a \in A\}$.
Полагаем $\mathrm{Sp}(W)$ подпространство V, порожденное W, $\langle A \rangle$– подмоноид, порожденный A в $\mathrm{End}(V)$.
Предполагаем, что элементы поля $\mathbb{F}$ заданы в некоторой конструктивной форме, причем определен размер задания.
Операции в $\mathbb{F}$ эффективны, представляются за полиномиальное время от размеров нормальных форм.

Основная лемма

Для $\alpha \in \mathbb{F}$ через $|\alpha|$ обозначается его размер.
Для $v = (\alpha_1, \ldots, \alpha_r) \in V$ полагаем
$|v| = \max |\alpha_i|$.
Для матрицы $a = (\alpha_{ij}) \in \mathrm{End}(V)$ полагаем
$|a| = \max\{|\alpha_{ij}|\}$.

Lemma (Основная лемма)

Существует алгоритм нахождения для данных конечных подмножеств $W \subseteq V$ и $U \subseteq End(V)$ базиса подпространства $Sp(W^{\langle U \rangle})$ в виде $w_1^{a_1}, \ldots, w_t^{a_t}$, где $w_i \in W$ и a_i – произведение элементов из U. Число использованных операций над элементами поля полиномиально по $r = \dim_{\mathbb{F}} V$ и количествам элементов W и U.

Доказательство основной леммы

Алгоритм:

1. Методом исключений Гаусса находим максимальное линейно независимое (л.н.) подмножество L_0 of W. Заметим, что $Sp(L_0^{\langle U \rangle}) = Sp(W^{\langle U \rangle})$.
2. Добавляем к множеству L_0 элементы $v^a, v \in L_0, a \in U$, проверяя каждый раз л. н. полученного множества. Таким образом будет построено максимальное л. н. подмножество L_1 множества $L_0 \cup L_0^U$ расширяющее L_0. Заметим, что $Sp(L_0^{\langle U \rangle}) = Sp(L_1^{\langle U \rangle})$, и элементы в L_1 имеют форму w или w^a, где $w \in W$ и $a \in U$. Отсюда, если $L_0 = L_1$, то L_0 – базис в $Sp(W^{\langle U \rangle})$.
3. Если $L_0 \neq L_1$, то повторяем процедуру для $L_1 \setminus L_0$ и находим максимальное л. н. подмножество L_2 в $L_1 \cup (L_1 \setminus L_0)^U$, расширяющее L_1. Строим $L_0 < L_1 < \ldots < L_i$ в V. Так как размерность r пространства V конечна, последовательность стабилизируется на $i \leq r$. Тогда L_i – базис в $Sp(W^{\langle U \rangle})$ из элементов требуемого вида.

Оценка сложности

Число операций в методе исключений Гаусса относительно матрицы размера $n \times r$ есть $O(n^2 r)$. Следовательно, требуется не более $O(n^2 r)$ шагов построения L_0 из W, где $n = |W|$ – число элементов W. Заметим, что $|L_j| \leq r$ для любого j. Поэтому нахождение L_{j+1} использует матрицу соответствующую $L_j \cup L_j^U$ размера не больше $r + r|U|$. Верхняя граница: $O(r^3|U|^2)$. Так как итераций $\leq r$, общая оценка $O(r^3|U|^2 + r|W|^2)$.

Corollary

При сделанных предположениях относительно $\mathbb{F}$ алгоритм основной леммы работает за полиномиальное от размеров $r = \dim_{\mathbb{F}} V$, $|W|$, $|U|$, и $\max\{|w|, |u| \mid w \in W, u \in U\}$ входа время.

Базовая модель

Пусть U_1 и U_2 – конечные подмножества полугруппы эндоморфизмов $\mathrm{End}(V)$; $\forall u_1 \in U_1, u_2 \in U_2 : u_1 u_2 = u_2 u_1$;
$A =< U_1 >, B =< U_2 >, v \in V, a \in A, b \in B$;
$\{a \in A, b \in B\}$ – секретные данные,
$\{U_1, U_2, v, v^a, v^b\}$ – открытые данные,

Theorem

По данным U_1, U_2, v, v^a, v^b за полиномиальное время находится вектор $v^{ab} = v^{ba}$.

Без вычисления a или b!

Алгоритм

1. По U_1 и v, используя алгоритм основной леммы (см. также следствие из основной леммы), за полиномиальное время находим базис $v^{a_1}, \ldots, v^{a_t}$, $a_i \in A$, пространства $Sp(v^A)$. Методом исключения Гаусса разлагаем v^a по этому базису:

$$v^a = \Sigma_{i=1}^{t} \alpha_i v^{a_i}, \quad \alpha_i \in \mathbb{F}.$$

2. Находим v^{ab} :

$$v^{ab} = (v^a)^b = (\Sigma_{i=1}^{t} \alpha_i v^{a_i})^b =$$

$$\Sigma_{i=1}^{t} \alpha_i v^{a_i b} = \Sigma_{i=1}^{t} \alpha_i v^{b a_i} = \Sigma_{i=1}^{t} \alpha_i (v^b)^{a_i}.$$

Нет необходимости в нахождении ни a, ни b, чтобы вычислить v^{ab}.

Заметим также, что нет необходимости знать U_2, достаточно того, что для некоторого $b \in \mathrm{End}(V)$ имеем $\forall (u \in U_1) ub = bu$.

The speaker and Myasnikov's cryptanalysis of Wang et. al. protocol

We propose new provable practical deterministic polynomial time algorithm for the braid Wang, Xu, Li, Lin and Wang Double shielded public key cryptosystems. We show that a linear decomposition attack based on the decomposition method introduced by the author works for the image of braids under the Lawrence-Krammer representation by finding the exchanging keys in the both two main protocols proposed in [X. Wang, C. Xu, G. Li, H. Lin and W. Wang, Double shielded public key cryptosystems, Cryptology ePrint Archive: Report 2014/558].

Wang et al. protocol

Wang et al. protocol:

- Public data: group G, element $h \in G$, and two subgroups $A = \mathrm{gp}(a_1, ..., a_n)$, $B = \mathrm{gp}(b_1, ..., b_m)$ of G, such that $[A, B] = 1$.
- Alice chooses four elements $c_1, c_2, d_1, d_2 \in A$, computes $x = d_1 c_1 h c_2 d_2$, and then sends x to Bob.
- Bob chooses six elements $f_1, f_2, g_1, g_2, g_3, g_4 \in B$, computes $y = g_1 f_1 h f_2 g_2$ and $w = g_3 f_1 x f_2 g_4$, and then sends (y, w) to Alice.
- Alice chooses two elements $d_3, d_4 \in A$, computes $z = d_3 c_1 y c_2 d_4$ and $u = d_1^{-1} w d_2^{-1}$, and then sends (z, u) to Bob.
- Bob sends $v = g_1^{-1} z g_2^{-1}$ to Alice.
- Alice computes $K_A = d_3^{-1} v d_4^{-1}$.
- Bob computes $K_B = g_3^{-1} u g_4^{-1} = c_1 f_1 h f_2 c_2$ which is equal to K_A and then $K = K_A = K_B$ is Alice and Bob's common secret key.

A cryptanalysis of Wang et. al. protocol

Now we show how the common secret key can be computed.
Let BzB be subspace of V generated by all elements of the form fzg where $f, g \in B$. We can construct a basis $\{e_i z l_i : e_i, l_i \in B\}$ of it in a polynomial time. Since $v \in BzB$, we can write it in the form

$$v = \sum_{i=1}^{r} \alpha_i e_i z l_i, \alpha_i \in \mathbb{F}. \tag{1}$$

Also we construct bases $\{e'_j h l'_j : e'_j, l'_j \in B\}$ and $\{e''_k w l''_k : e''_k, l''_k \in B\}$ of BwB. Then

$$y = \sum_{j=1}^{s} \beta_j e'_j h l'_j, \beta_j \in \mathbb{F}, \tag{2}$$

$$x = \sum_{k=1}^{q} \gamma_k e''_k w l''_k, \gamma_k \in \mathbb{F}. \tag{3}$$

A cryptanalysis of Wang et. al. protocol: revealing of a secret

Now we swap w by u in the right hand side of (3), and obtain

$$\sum_{k=1}^{q} \gamma_k e''_k u l''_k = \sum_{k=1}^{q} \gamma_k e''_k d_1^{-1} w d_2^{-1} l''_k =$$

$$d_1^{-1}(\sum_{k=1}^{q} \gamma_k e''_k w l''_k) d_2^{-1} = d_1^{-1} x d_2^{-1} = c_1 h c_2.$$

Then we swap h by $c_1 h c_2$ in the right hand side of (2).

$$\sum_{j=1}^{s} \beta_j f''_j c_1 h c_2 g''_j = c_1(\sum_{j=1}^{s} \beta_j e'_j h l'_j) c_2 = c_1 y c_2 = c_1 g_1 f_1 h f_2 g_2 c_2.$$

At last we swap z by $c_1 g_1 f_1 h f_2 g_2 c_2$ in the right hand side of (1) and get

$$\sum_{i=1}^{r} \alpha_i e_i c_1 g_1 f_1 h f_2 g_2 c_2 l_i = d_3^{-1}(\sum_{i=1}^{r} \alpha_i e_i z l_i) d_4^{-1} = c_1 f_1 h f_2 c_2 = K.$$

Algebraic versions: general scheme

Предлагается общая схема. Пусть G – некоторая алгебраическая система (группа, кольцо, лупа и т.п.). Выделяются две конечно порождённые полугруппы операторов A и B, действующих на G. Часто требуется, чтобы любой оператор $\alpha \in A$ был перестановочен с любым оператором $\beta \in B$. В ходе работы протокола публикуются данные относительно действия операторов на элементы G. Корреспонденты Алиса и Боб на основе этих данных и выбранных ими самими секретных операторов могут восстановить какой-то элемент из G. Криптостойкость алгоритма заключается в конечном итоге от невозможности реально найти этот результат постороннему наблюдателю, не владеющему секретами.

Криптоанализ

Метод линейного разложения и основанная на нём атака позволяют при условии, что G вложена каким-то эффективным образом в конечно мерное линейное пространство V, а операторы естественно продолжаются до эндоморфизмов V, раскрывать результат, не решая соответствующих алгоритмических задач поиска использованных ключей.

A new function on groups

Definition

Let $\mathcal{K}$ be a class of finitely generated groups. Function $\rho : \mathcal{K} \to \mathbb{N} \cup \{\infty\}$ is defined as follows. Let $G \in \mathcal{K}$ be a group in $\mathcal{K}$ equipped with a fixed finite generating set X. Then:

- For each $g \in G$ we set $\rho_X(g) = l$, where l is the minimal number such that the normal closure $ncl(g)$ of g is generated by elements of the form g^f, where the word length of f w.r.t. X is $\leq l$.
- We set $\rho_X(G) = max\{\rho_X(g) : g \in G\}$.

The function $\rho(G)$ is defined as the equivalence class of functions $\rho_X(G)$ for all possible X.

Nilpotent case

It clear that $\rho(A) = 0$ on every abelian group A.

Theorem

Let G be a finitely generated nilpotent group of class c. Then $\rho(G) \leq c - 1$.

Problem

Give an upper bound of $\rho(G)$ for an arbitrary polycyclic group G.

An example of key transmitted scheme

Let G be an algebraic structure and g be a distinguished element of G Let A and B be two elementwise permutable subsemigroups of End$G$.

- Alice chooses $\alpha \in A$ and public $\alpha(g)$.
- Bob chooses $\beta \in B$ and publics $\beta(g)$.
- The transmitted key: $K = \alpha(\beta(g)) = \beta(\alpha(g))$.

A non linear attack

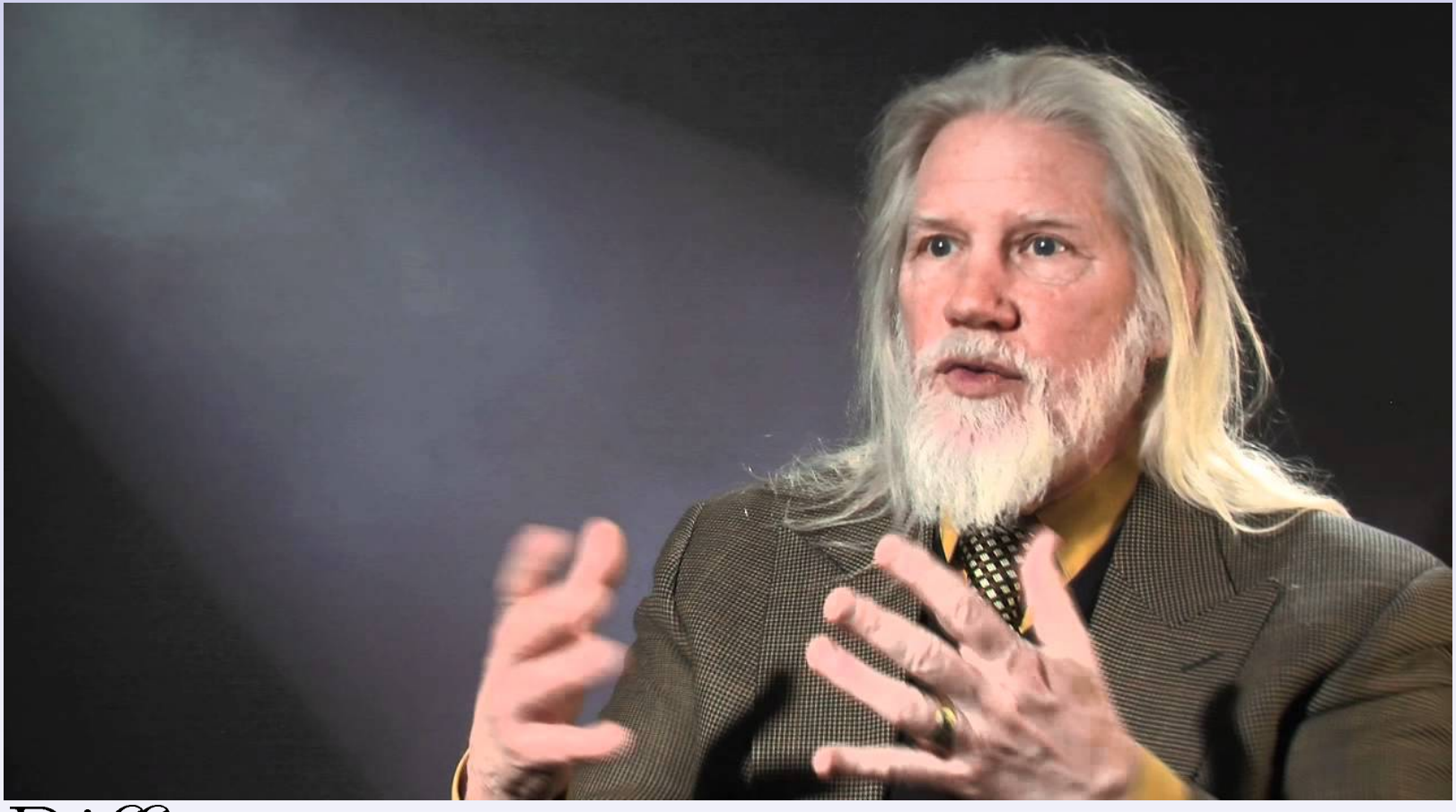
We compute a set X of generating elements of a subsystem, generated by all elements of the form $\lambda(g), \lambda \in A$. It is off-line procedure. Let $X = \{\lambda_i(g) : i = 1, ..., t\}$.
Then we write $\alpha(g)$ as an expression of the form

$$\alpha(g) = w(\lambda_1(g), ..., \lambda_t(g)),$$

where w is a term.
Then we swap g with $\beta(g)$, then we get

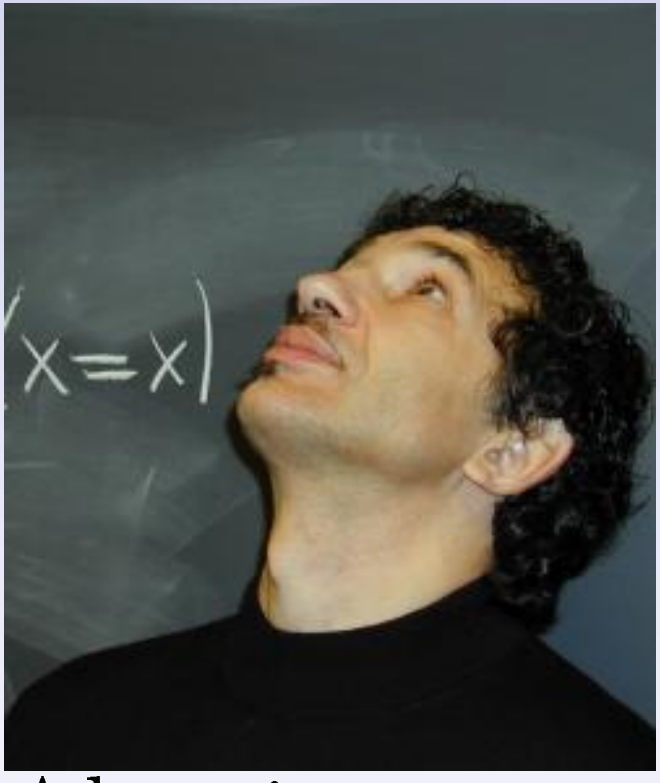
$$w(\lambda_1(\beta(g)), ..., \lambda_t(\beta(g)) = \beta(w(\lambda_1(g), ..., \lambda_t(g)) = \beta(\alpha(g)) = K.$$

Crypto Gallery



Diffie

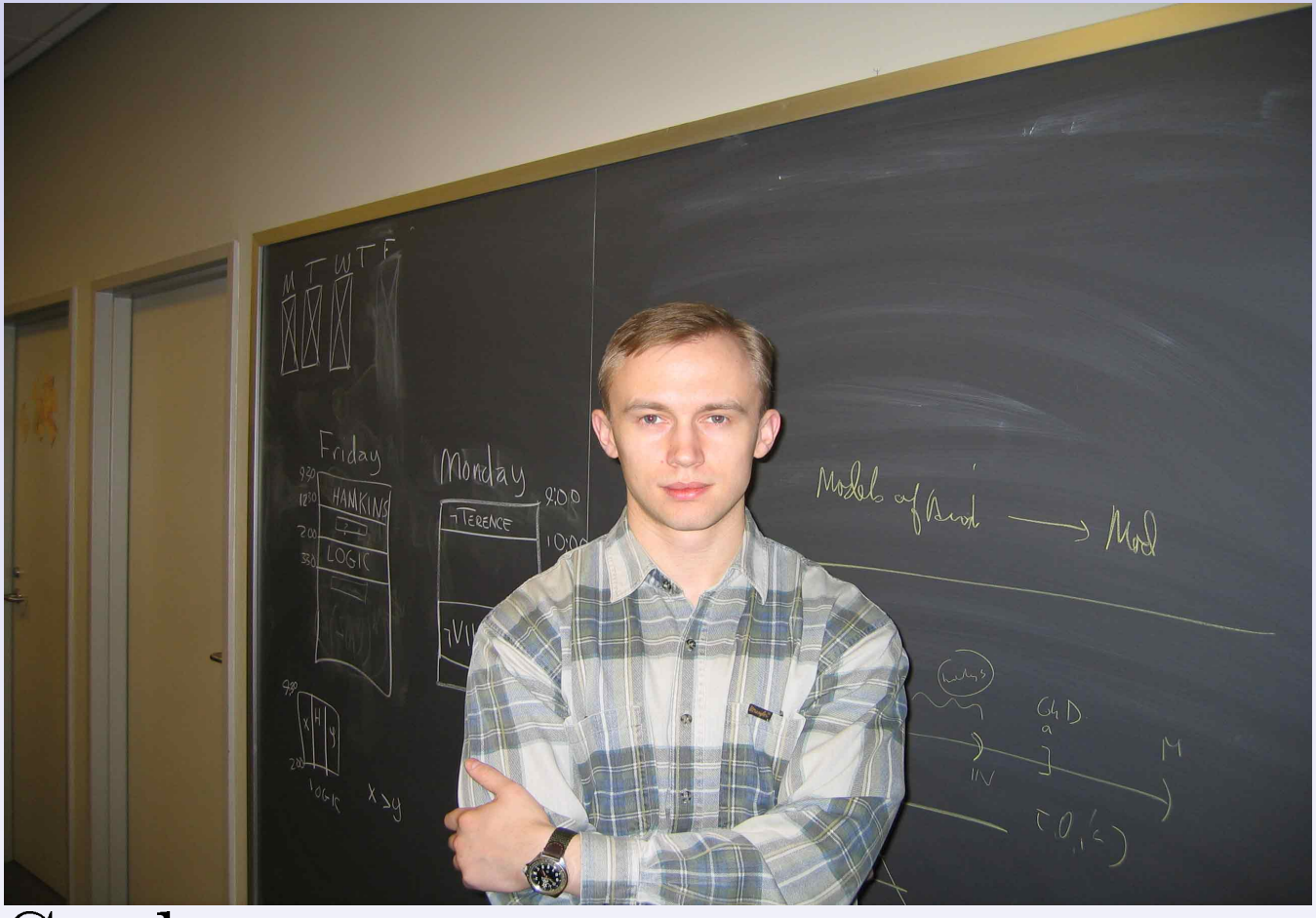
Delaram and Gilbert



Alexei

Vladimir

Ushakov



Sasha

Roman'kov and Jarkova

Vitaly and Nastya

The end

СПАСИБО!
THANK YOU!
ESTOY AGRADECIDO!