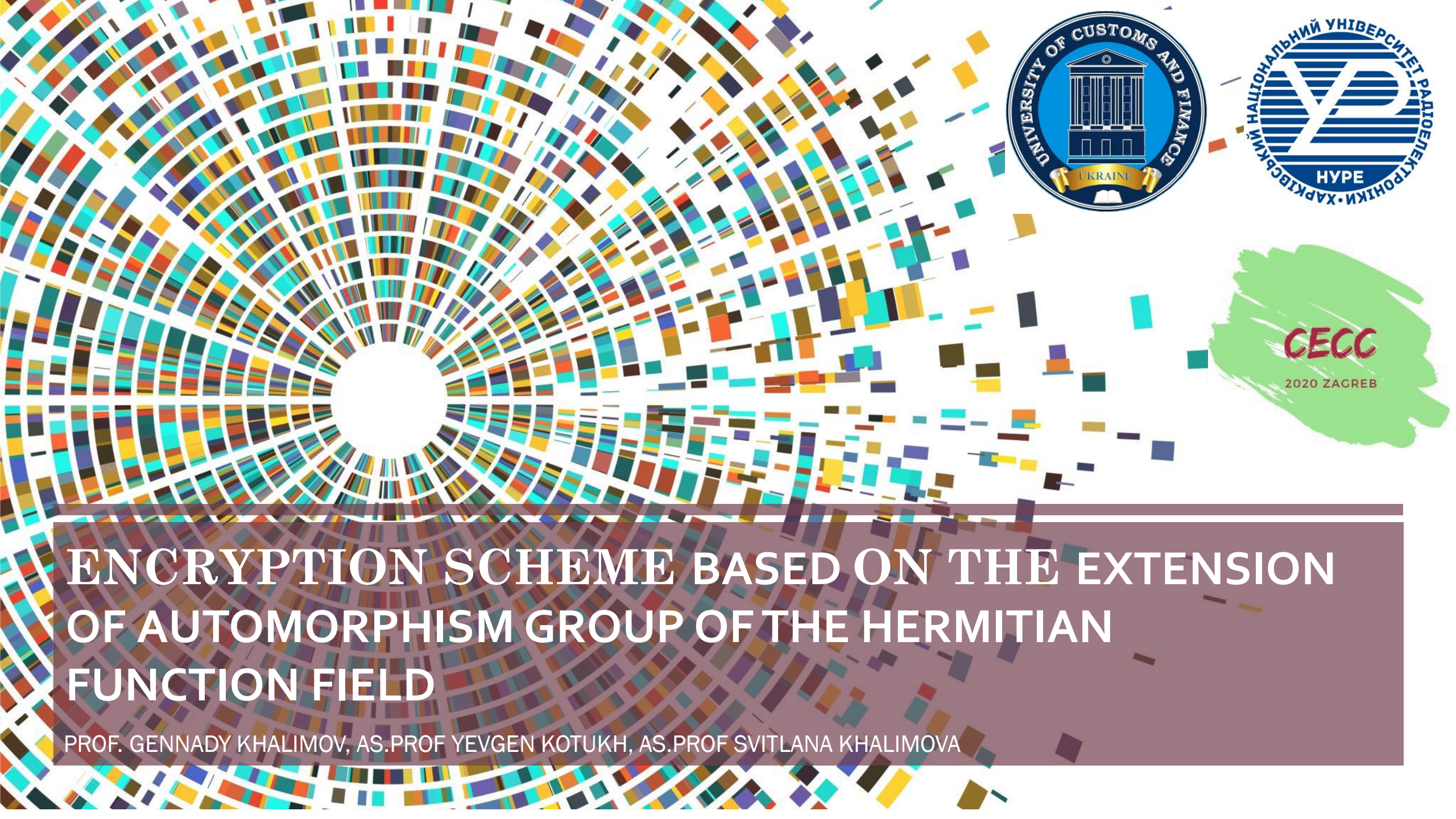

ENCRYPTION SCHEME BASED ON THE EXTENSION OF AUTOMORPHISM GROUP OF THE HERMITIAN FUNCTION FIELD

PROF. GENNADY KHALIMOV, AS.PROF YEVGEN KOTUKH, AS.PROF SVITLANA KHALIMOVA

Outline

1. Era of Post-Quantum Primitives
2. MST idea/MST3 present results
3. Proposed solution
4. Key Generation
5. Encryption
6. Decryption
7. Security Analysis
8. Conclusion

Era of Post-Quantum Approaches

Name	Type	Basis	Private key, bit	Public key, bit	Cipher text/signature
Lepton	Encryption	Code	80	4128	5557
3Bears	Encryption, KEM	Code	40	1584	1697
qTESLA	Signature	Code	4128	6432	5920
Classic McElliece	Encryption	Code	13908	1047319	22
LEDAcrypt	Encryption, KEM	Code	40	18016	9008
Dilithium	Signature	Code	3856	760	3366
FrodoKEM	KEM	Lattice	31272	15632	15768
RQC	KEM	Lattice	3510	3510	3574
NTRU	Encryption	Lattice	6230	6734	140
SIKE	KEM	Isogeny	826	726	766
Rainbow	Signature	Multivariant Polynom	1319000	871000	118
LUOV	Signature	Multivariant Polynom	32	39300	4700
SPHINCS	Signature	Hash	1024	1024	41800
Picnic	Signature	Other	256	512	209474
RVB	KEM	Other	334	332	
WalnutDSA	Signature	Other	1040	634	7704
pqRSA	Encryption, Signature	Other	25769803776	8589934592	8589934592
RSA	Encryption, Signature, KEM	FF-DLP	3/384	384	384
Elliptic curves	Encryption, Signature, KEM	DLP	64	128	128

Try to satisfy: Encryption. KED and DES

To have less computation complexity and faster realization

Less cipher/signature data

The MST Cryptosystem

Requirements for cryptosystems on group algebra	- to use a logarithmic signatures and coverages as a type of factorization of a finite group - to use a large group with a large non-trivial center
Review	[3] S.S. Magliveras, "A cryptosystem from logarithmic signatures of finite groups", in Proceedings of the 29th Midwest Symposium on Circuits and Systems , pp. 972–975, Elsevier Publishing, Amsterdam, The Netherlands, 1986. [4] W. Lempken, S.S. Magliveras, Tran van Trung and W. Wei, "A public key cryptosystem based on non-abelian finite groups", J. of Cryptology, 22 (2009), 62–74. [5] S.S.Magliveras, D.R.Stinson, and T.vanTrung, "New approaches to designing public key cryptosystems using one-way functions and trapdoors in finite groups", Journal of Cryptology , vol. 15, no. 4, pp. 285–297, 2002. [6] S.S. Magliveras, P. Svaba, T. Van Trung, and P. Zajac, "On the security of a realization of cryptosystem MST3", Tatra Mountains Mathematical Publications ,vol.41,pp.65–78,2008.
Definition1 (cover (logarithmic signature) mappings)	Let $\alpha = [A_1,...,A_s]$ be a cover (logarithmic signature) of type $(r_1,r_2,...,r_s)$ for G with $A_i = [a_{i,1},a_{i,2},...,a_{i,r_i}]$, where $m = \prod_{i=1}^{s} r_i$. Let $m_1 = 1$ and $m_i = \prod_{j=1}^{i-1} r_j$ for $i = 2,...,s$. Let τ denote the canonical bijection $\tau : \mathbb{Z}_{r_1} \times \mathbb{Z}_{r_2} \times ... \times \mathbb{Z}_{r_s} \to \mathbb{Z}_m, \quad \tau(j_1,j_2,...,j_s) = \sum_{i=1}^{s} j_i \cdot m_i$. Then the surjective (bijection) mapping $\alpha' : \mathbb{Z}_m \to G$ induced by is $\alpha'(x) = a_{1j_1} \cdot a_{2j_2} \cdots a_{sj_s}$ where $(j_1,j_2,...,j_s) = \tau^{-1}(x)$. More generally, if $\alpha = [A_1,...,A_s]$ is a logarithmic signature (cover) for, then each element $g \in G$ can be expressed uniquely (at least one way) as a product of the form $g = a_1 \cdot a_2 \cdots a_s$, for $a_i \in A_i$

As lager order than stronger cryptography

Description of MST3 algorithm

Initial setting	Let G be a finite non-abelian group with nontrivial center Z
Generate	a tame logarithmic signatures $\beta = [B_1, B_2, ..., B_s] := (b_{ij})$ the class $(r_1, r_2, ..., r_s)$ for $\mathbb{Z}$ - a random cover $\alpha = [A_1, A_2, ..., A_s] := (a_{i,j})$ the same class as β - a set of elements $t_0, t_1 ..., t_s \in G \setminus \mathbb{Z}$;
Definition	of homomorphism to calculate $f : G \to \mathbb{Z}$;
Calculate	$\gamma := (h_{ij}) = (t_{i-1}^{-1} f(a_{ij}) b_{ij} t_i)$ for $i = 1,...,s$, $j = 1,...,r_i$.
Output: public key private key	$(\alpha = (a_{ij}), \gamma = (h_{ij}), f)$ $(\beta = (b_{ij}), (t_0, ..., t_s))$
Encryption	
Input the message to be encrypted. set a random number	$x \in \mathbb{Z}_{\lvert Z \rvert}$ $R \in Z_{\lvert \mathbb{Z} \rvert}$
Calculate	$y_1 = \alpha'(R) \cdot x$, $y_2 = \gamma(R) = t_0^{-1} f(\alpha'(R)) b'(R) t_s$.
Transmit	$y = (y_1, y_2)$
Decryption	
Input the cipher text the private key the function of the homomorphism	$y = (y_1, y_2)$, $(\beta = (b_{ij}), (t_0, ..., t_s))$ $f : G \to \mathbb{Z}$.
Calculate	$\beta(R) = y_2 t_s^{-1} f(y_1)^{-1} t_0$.
Recover	R with $\beta(R)$ using β^{-1}
Calculate	$x = \alpha'(R)^{-1} \cdot y_1$

Alpha to be used for Public key,
Beta and t – for private

Remark. Message x masked by a logarithmic signature on the arrays $\alpha = (a_{ij})$, $\gamma = (h_{ij})$ which is calculated for a random number R .

The function of the homomorphism $f : G \to \mathbb{Z}$ moves the group element to the center of the group.

Calculation $\beta(R) = y_2 t_s^{-1} f(y_1)^{-1} t_0$ and the subsequent recovery of R is possible due to the commutativity of the center.

MST3 results for Suzuki groups

Definition Suzuki groups	$S(a,b)=\begin{pmatrix}1 & 0 & 0\\ a & 1 & 0\\ b & a^{\theta} & 1\end{pmatrix}$, $a,b\in F_q$, $q=2^m$, $3\le m\in N$
The center	$Z\left(S(a,b)\right)=\left\{S(0,b)\middle\vert b\in F_q\right\}$
Multiplication operation	$S(a_1,b_1)S(a_2,b_2)=S(a_1+a_2,b_1+b_2+a_1^{\theta}a_2)$
Search of inverse element	$S(a,b)^{-1}=S(a,b+a^{\theta+1})$
Experimental Results	Encryption testing was performed on a computer running Ubuntu 16.04, Intel® Core ™ i7-4702MQ CPU @ 2.20 GHz processor, 12GB of RAM, the results are presented in Table 1.

Suzuki group has a big center and simple group operation

The secrecy of the cryptosystem is defined by the finite field for determining the coordinates of the Suzuki group elements. For 256 bits-like security, the field should be 2 ^256.

Table 1. Expenses for encryption / decryption

	Expenses for encryption/decryption in the finite field 128 bits. Breakdown classes			Expenses for encryption / decryption by RSA algorithm. Bit rate of key parameters, bit			
	64[4]	32[16]	16[256]	512	1024	2048	4096
Time to generate key data, ms	56	59	169	3,368	8,685	63,65	707,6
The size of a private key, byte	78830	111726	671918	342	632	1214	2373
The size of the public key, byte	39761	75217	590609	92	160	292	548
Encryption time 100 KB, ms	4749	2388	1205	66,98	117,9	243,8	591,8
Time of decoding 100 KB, ms	2711	1487	888	641,27	2116,4	9853,5	64250,4

When calculating in the final field 2048 and 4096 bits, the encryption and decryption time of the RSA algorithm is tens of times larger compared to the cryptosystem, but provides significant cost savings for the size of private and public keys.

We are interested in groups with large order; potentially, for a smaller field, we can achieve less computational complexity without reducing security.

The automorphism group of the Hermitian function field

Definition	The Hermitian function field $H\left\|F_{q^2}\right.$ generated by two elements x, y of $y^q + y = x^{q+1}$ The automorphism group of the Hermitian function field $A := Aut(H) = \{\sigma : H \mapsto H \mid \sigma$ is isomorphic to the projective unitary group $PGU\left(3, F_{q^2}\right)$ and has order $ordA = q^3\left(q^2-1\right)\left(q^3+1\right)$.	
Review	[1] H.Stichtenoth, "Über die Automorphismengruppe eines algebraischen Funktionenkörpers von Primzahlcharakteristik" I, II, Arch. Math. 24, pp.524–544 and pp.615–631, 1973. [2] A. Garcia, H. Stichtenoth, C.-P.Xing, "On Subfields of the Hermitian Function Field", Kluwer Academic Publishers, Compositio Mathematica 120: pp.137–170, 2000.	
Group $A(P_\infty)$ P_∞ the unique place in H over the pole of x.	All automorphisms of $H\left\|F_{q^2}\right.$ acting on it as $\begin{cases}\sigma(x) = ax+b \\ \sigma(y) = a^{q+1}y + ab^q x + c,\end{cases}$ $\left\{a \in F_{q^2}^* := F_{q^2}\setminus\{0\},\quad b \in F_{q^2},\quad c^q + c = b^{q+1}\right\}$ $ordA(P_\infty) = q^3\left(q^2-1\right)$.	
Properties the group structure	Group operation $[a_1,b_1,c_1]\cdot[a_2,b_2,c_2] = \left[a_1a_2, a_2b_1+b_2, a_2^{q+1}c_1 + a_2b_2^q b_1 + c_2\right]$ The identity is the triple $[1,0,0]$ The inverse of $[a,b,c]$ is $[a,b,c]^{-1} = \left[a^{-1}, -a^{-1}b, a^{-(q+1)}c^q\right]$ For the characteristic is odd, we can represent $A(P_\infty)$ as follows: $A(P_\infty)\left\{\left[a, b, \frac{b^{q+1}}{2}+c\right] \middle\| a \in F_{q^2}^*, b \in F_{q^2} \quad and \quad c^q + c = 0\right\}$.	
Another automorphism	$A_1(P_\infty) = \{\sigma \in A(P_\infty) \mid \sigma(x) = x+b$	$\sigma(x) = x+b, \quad \sigma(y) = y + b^q x + c$
	The factor group $A(P_\infty)/A_1(P_\infty)$	$\partial(x) = \alpha x, \quad \partial(y) = a^{q+1}y$
		$\omega(x) = x/y, \quad \omega(y) = 1/y$

Max order equals q^8

Many automorphism are exist. Automorphism with a group operation for sigma mappings was chosen for the cryptosystem implementation

The calculations are greatly simplified to determine the group in the field of odd characteristics. The three coordinate representation of the elements of the group is also simplified. There is no need to solve the H equation for the coordinate C.

The automorphism group $A(P_\infty)$ **has a** $ordA(P_\infty) = q^3\left(q^2-1\right)$ **greater** than the order Suzuki group.

A larger group order gives potential an advantage to cryptosystem secrecy and computing over small fields has an advantage in implementation.

Encryption scheme on the extension of automorphism group of the Hermitian function field

Input a large group on the field of odd characteristic	$A(P_\infty)=\{S(a,b,c)\mid\ a\in F_{q^2}^{*}:=F_{q^2}\setminus\{0\}, b\in F_{q^2}, c^q+c=0\}$
Choose a tame logarithmic signature	$\beta_{(1)}=\left[B_{1(1)},\ldots,B_{s(1)}\right]=\left(b_{ij}\right)_{(1)}=S\left(1,b_{ij(1)},b_{ij(1)}^{q+1}/2\right)$ of type $\left(r_{1(1)},\ldots,r_{s(1)}\right)$, $i=\overline{1,s(1)}$, $j=\overline{1,r_{i(1)}}$, $b_{ij(1)}\in F_{q^2}$. $\beta_{(2)}=\left[B_{1(2)},\ldots,B_{s(2)}\right]=\left(b_{ij}\right)_{(2)}=S\left(1,0,b_{ij(2)}\right)$ of type $\left(r_{1(2)},\ldots,r_{s(2)}\right)$, $i=\overline{1,s(2)}$, $j=\overline{1,r_{i(2)}}$, $b_{ij(2)}\in F_q\subset F_{q^2}$.
Select a random cover	$\alpha_{(1)}=\left[A_{1(1)},\ldots,A_{s(1)}\right]=\left(a_{ij}\right)_{(1)}=S\left(a_{ij(1)_1},a_{ij(1)_2},\left(a_{ij(1)_2}\right)^{q+1}/2+a_{ij(1)_3}\right)$ of the same type as $\beta_{(1)}$, where $a_{ij}\in A(P_\infty)$, $a_{ij(1)_1},a_{ij(1)_2}\in F_{q^2}\setminus\{0\}$. $\alpha_{(2)}=\left[A_{1(2)},\ldots,A_{s(2)}\right]=\left(a_{ij}\right)_{(2)}=S\left(a_{ij(2)_1},a_{ij(2)_2},\left(a_{ij(2)_2}\right)^{q+1}/2+a_{ij(2)_3}\right)$ of the same type as $\beta_{(2)}$, where $a_{ij(2)_1},a_{ij(2)_2},a_{ij(2)_3}\in F_q\setminus\{0\}\subset F_{q^2}$.
Choose $t_{0(k)},t_{1(k)},\ldots,t_{s(k)}\in A(P_\infty)\setminus Z$,	$t_{i(k)}=S\left(t_{i(k)_1},t_{i(k)_2},(t_{i(k)_2})^{q+1}/2\right), t_{i(k)_j}\in F^{\times}, i=\overline{0,s(k)}, j=\overline{1,2}, k=\overline{1,2}$. Let's $t_{s(1)}=t_{0(2)}$.
Construct a homomorphism	$f_1\left(S\left(a_1,a_2,a_2^{q+1}/2\right)\right)=S\left(1,a_2,a_2^{q+1}/2\right)$ $f_2\left(S\left(a_1,a_2,a_2^{q+1}/2\right)\right)=S\left(1,0,a_2\right)$
Compute	$\gamma_{(1)}=\left[h_{1(1)},\ldots,h_{s(1)}\right]=\left(h_{ij}\right)_{(1)}=t_{(i-1)(1)}^{-1}f_1\left(\left(a_{ij}\right)_{(1)}\right)\left(b_{ij}\right)_{(1)}t_{i(1)}, i=\overline{1,s(1)}, j=\overline{1,r_{i(1)}}$ $\gamma_{(2)}=\left[h_{1(2)},\ldots,h_{s(2)}\right]=\left(h_{ij}\right)_{(2)}=t_{(i-1)(2)}^{-1}f_2\left(\left(a_{ij}\right)_{(2)}\right)\left(b_{ij}\right)_{(2)}t_{i(2)}, i=\overline{1,s(2)}, j=\overline{1,r_{i(2)}}$
Output: public key	$[f_1,f_2,(\alpha_k,\gamma_k)]$,
private key	$\left[\beta_{(k)},\left(t_{0(k)},\ldots,t_{s(k)}\right)\right], k=\overline{1,2}$.

Our proposal for improvement is determined by the fact that in addition to the logarithmic signature in the center of the group, we added another logarithmic signature along the coordinate B.

Thus, we have increased the number of arrays of Beta, noise, Alpha, Gamma, and t accordingly.

Example 1 Construction of a tame logarithmic signature

Example 1 Construction of a simple logarithmic signature $\beta = [B_1,...,B_s] = (b_{ij})$

Fix the finite field F_q, $q = 2^{10}$, type $(r_1,...,r_s) = (2^2,2^2,2^3,2^3)$.

initial installation						adding noise						fusion $C_1 = (B_1,B_4)$, $C_2 = B_2$, $C_3 = B_3$					
$\beta(1)=$	B_1	00	00	000	000	$\beta(2)=$	B_1	00	00	000	000	$\beta(3)=$	C_1	**01**	**00**	**110**	000
		10	00	000	000			10	00	000	000			**00**	**11**	**010**	100
		01	00	000	000			01	00	000	000			**11**	**00**	**011**	010
		11	00	000	000			11	00	000	000			**10**	**01**	**000**	110
	B_2	00	00	000	000		B_2	**11**	00	000	□00			**01**	**10**	**101**	001
		00	10	000	000			**10**	10	000	000			**01**	**10**	**010**	101
		00	01	000	000			**10**	01	000	000			**00**	**11**	**100**	011
		00	11	000	000			**01**	11	000	000			.	.	.	.
	B_3	00	00	000	000		B_3	**10**	**10**	000	000			.	.	.	.
		00	00	100	000			**01**	**11**	100	000			**10**	**10**	**010**	101
		00	00	010	000			**01**	**00**	010	000			**11**	**11**	**100**	011
		00	00	110	000			**00**	**10**	110	000			**00**	**01**	**001**	111
		00	00	001	000			**11**	**01**	001	000		C_2	**11**	00	000	000
		00	00	101	000			**10**	**01**	101	000			**10**	10	000	000
		00	0	011	000			**01**	**11**	011	000			**10**	01	000	000
		00	00	111	000			**00**	**10**	111	000			**0**	11	000	000
	B_4	00	00	000	000		B_4	**01**	**00**	**110**	000		C_3	**10**	**10**	000	000
		00	00	000	100			**00**	**11**	**010**	100			**01**	**11**	100	000
		00	00	000	010			**11**	**00**	**011**	010			**01**	**00**	010	000
		00	00	000	110			**10**	**01**	**000**	110			**00**	**10**	110	000
		00	00	000	001			**01**	**10**	**101**	001			**11**	**01**	001	000
		00	00	000	101			**01**	**10**	**010**	101			**10**	**01**	101	000
		00	00	000	011			**00**	**11**	**100**	011			**01**	**11**	011	000
		00	00	000	111			**11**	**01**	**001**	111			**00**	**10**	111	000

$R = 673 \quad R = (R_1,R_2,R_3) = (1,1,5) \quad R_1 + R_2 2^5 + R_3 2^7 = 673$

$\beta(R) = C_1(R_1) + C_2(R_2) + C_3(R_3) = 0011010100 + 1010000000 + 1001101000 = 0000111100$

$\beta(R')^{-1} = 10|00|010|101 = (R_1',R_2',R_3',R_4') = (0,1,5,1)$

$\beta(R')^{-1} := \mu\{(0,1,5,1)\} \to (1\|0,1,5) = (10000,10,101) \qquad R = (R_1,R_2,R_3) = (1,1,5)$

Example 2 Key Generation

Example 2 Key Generation

Fix the finite field F_q, $q = 3^4$, $g(x) = x^4 + x + 2$, types $(r_{1(1)}, r_{2(1)}) = (3^2, 3^2)$, $(r_{1(2)}, r_{2(2)}) = (3,3)$.

$\beta_{(k)} = [B_{1(k)}, B_{2(k)}] =$			$S(1, b_{ij(1)}, b_{ij(1)}^{q+1}/2)$	$\alpha_{(k)} = S(a_{ij(k)_1}, a_{ij(k)_2}, (a_{ij(k)_2})^{q+1}/2)$			$t_{i(k)} = S(t_{i(k)_1}, t_{i(k)_2}, (t_{i(k)_2})^{q+1}/2)$			$\gamma_{(k)} = t^{-1}_{(i-1)(k)} f_k((a_{ij})_{(k)})(b_{ij})_{(k)} t_{i(k)}$		
$\beta_{(1)}=$	$B_{1(1)}$	00 00	a0 0 0	$\alpha_{(1)}=$	$A_{1(1)}$	a38 a66 a60	$t_{(1)}=$	$t_{0(1)}$	a27 a28 a0	$\gamma_{(1)}=$	$h_{1(1)}$	a79 a39 a53
		10 00	a0 a0 a40			a8 a17 a50		$t^{-1}{}_{0(1)}$	a53 a41 a50			a79 a31 a19
		20 00	a0 a40 a40			a35 a35 a70						a79 a78 a46
		01 00	a0 a1 a50			a39 a44 a0						a79 a43 a70
		11 00	a0 a53 a10			a68 a46 a20						a79 a2 a27
		21 00	a0 a44 a0			a16 a5 a10						a79 a66 a60
		02 00	a0 a41 a50			a48 a51 a70						a79 a64 a63
		12 00	a0 a4 a0			a12 a1 a50						a79 a34 a58
		22 00	a0 a13 a10			a78 a1 a50		$t_{1(1)}$	a26 a71 a30			a79 a5 a4
	$B_{2(1)}$	21 00	a0 a44 a0		$A_{2(1)}$	a28 a36 a0		$t^{-1}{}_{1(1)}$	a54 a5 a10		$h_{2(1)}$	a62 a71 a28
		12 10	a0 a26 a60			a20 a60 a0						a62 a11 a59
		02 20	a0 a14 a20			a52 a19 a70						a62 a50 a27
		12 01	a0 a56 a40			a9 a55 a30						a62 a28 a26
		01 11	a0 a9 a50			a16 a1 a50						a62 a76 a23
		20 21	a0 a32 a40			a5 a18 a60						a62 a41 a76
		20 02	a0 a39 a30			a79 a28 a0						a62 a76 a69
		11 12	a0 a69 a10			a11 a1 a50						a62 a28 a69
		11 22	a0 a30 a20			a43 a67 a70		$t_{2(1)}$	a8 a74 a60			a62 a68 a78
$\beta_{(2)}=$	$B_{1(2)}$	00 00	a0 0 0	$\alpha_{(2)}=$	$A_{1(2)}$	a61 a51 a70	$t_{(2)}=$	$t^{-1}{}_{0(2)}$	a72 a26 a60	$\gamma_{(2)}=$	$h_{1(2)}$	a31 a48 a68
		10 00	a0 0 a0			a5 a30 a20						a31 a48 a61
		20 00	a0 0 a40			a57 a44 a0		$t_{1(2)}$	a39 a32 a40			a31 a48 a48
	$B_{2(2)}$	00 00	a0 0 0		$A_{2(2)}$	a44 a75 a70		$t^{-1}{}_{1(2)}$	a41 a33 a50		$h_{2(2)}$	a45 a9 a39
		11 00	a0 0 a53			a21 a40 a40		$t^{-1}{}_{2(2)}$	a76 a72 a40			a45 a9 a22
		22 00	a0 0 a13			a0 a38 a20		$t_{2(2)}$	a4 a36 a0			a45 a9 a36

An example with constructed arrays for a finite field 3^4.

Two Beta arrays, two noisy alpha, gamma, t.

Encryption scheme on the extension of automorphism group of the Hermitian function field

Encryption

Input a message $m \in A(P_\infty)$ the public key.	$m = S(m_1, m_2, m_3)$, $m_1, m_2 \in F_{q^2}$, $m_3 \in F_q \subset F_{q^2}$ $[f_1, f_2, (\alpha_k, \gamma_k)]$ $k = \overline{1,2}$	m = [a1 a2 a3] R=(R_1,R_2)=(29,7), R_1=(2,3), R_2=(1,2)
Compute	$y_1 = \alpha'(R) \cdot m = \alpha_1'(R_1) \cdot \alpha_2'(R_2) \cdot m$	y_1 =[a50 a56 a38]
Compute	$y_2 = \gamma'(R) = \gamma_1'(R_1) \cdot \gamma_2'(R_2) =$ $S(*, a_{(1)_2}(R_1) + \beta_{(1)}(R_1) + *, a_{(2)_2}(R_2) + \beta_{(2)}(R_2) + *)$.	y_2=[a57 a36 a23]
Compute	$y_3 = f_1(\alpha_1'(R_1)) = S(1, a_{(1)_2}(R_1), *)$	y_3 =[a0 a65 a50]
Compute	$y_4 = f_2(\alpha_2'(R_2)) = S(1, 0, a_{(2)_2}(R_2))$	y_4 =[a0 0 a9]
Output:	(y_1, y_2, y_3, y_4).	

Decryption

Input a ciphertext the private key.	(y_1, y_2, y_3, y_4) $[\beta_{(k)}, (t_{0(k)}, ..., t_{s(k)})]$ $k = \overline{1,2}$	y_1 =[a50 a56 a38], y_2=[a57 a36 a23], y_3=[a0 a65 a50], y_4=[a0 0 a9]
Compute	$D^{(1)}(R_1, R_2) = t_{0(1)} \cdot y_2 t_{s(2)}^{-1} = S(1, a_{(1)_2}(R_1) + \beta_{(1)}(R_1),$ $a_{(2)_2}(R_2) + \beta_{(2)}(R_2) + (a_{(1)_2}(R_1) + \beta_{(1)}(R_1))^{q+1} / 2 + *)$.	$D^{(1)}(R_1, R_2)$ = [a27 a28 a0] [a57 a36 a23] [a76 a72 a40]=[a0 a1 a32]
Compute	$D^*(R) = y_3^{-1} D^{(1)}(R_1, R_2) =$ $S(1, \beta_{(1)}(R_1), a_{(2)_2}(R_2) + \beta_{(2)}(R_2) + *)$.	$D^*(R)$ = [a0 a65 a50]$^{-1}$[a0 a1 a32]= [a0 a25 a50] [a0 a1 a32]=[a0 a18 a42]
Restore R_1	$\beta_{(1)}(R_1)^{-1}$	R_1:→a18=0201=2000+1201→(2,3)
Compute	$y_2^{(1)} = \gamma_1'(R_1)^{-1} y_2 = \gamma_2'(R_2) =$ $S(*, *, a_{(2)_2}(R_2) + \beta_{(2)}(R_2) + *)$.	$y_2^{(1)}$ = ([a79 a78 a46][a62 a28 a26])$^{-1}$[a57 a36 a23]= [a19 a53 a72] [a57 a36 a23]= [a76 a22 a46]
Compute	$D^{(2)}(R_2) = t_{0(2)} \cdot y_2 t_{s(2)}^{-1} = S(1, 0, a_{(2)_2}(R_2) + \beta_{(2)}(R_2))$	$D^{(2)}(R_2)$ = [a8 a74 a60] [a76 a22 a46] [a76 a72 a40]=[a0 0 a55]
Compute	$D^*(R) = y_4^{-1} D^{(2)}(R_2) = S(1, 0, \beta_{(2)}(R_2))$	$D^*(R)$ = [a0 0 a9] [a0 0 a55]= [a0 0 a41]
Restore R_2	$\beta_{(2)}(R_2)^{-1}$	R_2:→a41=0200= 1000+2200→(1,2)
Output:	$R = (R_1, R_2)$ $m = \alpha'(R_1, R_2)^{-1} \cdot y_1$	m = ([a35 a35 a70][a9 a55 a30][a5 a30 a20][a0 a38 a20])$^{-1}$ [a50 a56 a38]= [a49 a60 a47]$^{-1}$[a50 a56 a38]= [a31 a51 a13] [a50 a56 a38]=[a1 a2 a3]

Two session keys for the encryption to be used (same type as log signatures)

Output will be 2 times bigger if 3rd coordinate will be used

Decryption to be executed in two iterations. Each iteration recovers R session keys.

Security Analysis

Security Analysis

type of attack	*attack mechanism*	*complexity*	*Possible solution*
brute force attack on cipher text	selection $R=(R_1,R_2)$ for $y_1=\alpha'(R)\cdot m=\alpha_1'(R_1)\cdot\alpha_2'(R_2)\cdot m$	q^3	
brute force attack on $R=(R_1,R_2)$	selection $R=(R_1,R_2)$ of $y_2=\gamma'(R)=\gamma_1'(R_1)\cdot\gamma_2'(R_2)$	q^3	
	selection R_1 of $y_3=f_1(\alpha_1'(R_1))=S(1,a_{(1)_2}(R_1),*)$	q^2	link y_3 and y_4 through the product and matrix transformation *complexity--* q^3
	selection R_2 of $y_4=f_2(\alpha_2'(R_2))=S(1,0,a_{(2)_2}(R_2))$	q	
brute force attack on $(t_{0(k)},\ldots,t_{s(k)})$	selection $(t_{0(k)},\ldots,t_{s(k)})$	$(q^2)^3$	
attack on the algorithm	Extraction parameters $a_{(1)_2}(R_1)$, $a_{(2)_2}(R_2)$ of $y_3=S(1,a_{(1)_2}(R_1),*)$, $y_4=S(1,0,a_{(2)_2}(R_2))$ does not allow to calculate $\alpha_1'(R_1)\cdot\alpha_2'(R_2)$ in $y_1=\alpha_1'(R_1)\cdot\alpha_2'(R_2)\cdot m$		

Potential security defined by q^3 value. Session keys should be bind by matrix transformation

Conclusions

Pros	- high secrecy $\sim q^3$ for encryption scheme based on automorphism group of the Hermitian function field over F_q - length ciphertext is $3\log q$ for computing in the finite field over F_q - the computing (encryption time in particular) in the finite field are smaller compare to the cryptosystem in the Suzuki group - the length of the logarithmic signature array is determined by the finite field over F_q and significantly less compared to the Suzuki cryptosystem
Cons	- for decrypting, it is necessary to calculate the inverse element three times - the large key data size
future research	- construction of cryptosystems based on large-order groups

Thank you very much for your attention!

Gennady Khalimov

Yevgen Kotukh

Svetlana Khalimov