

Foundations of Adaptive Networked Societies of Tiny Artefacts

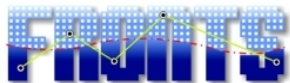
Security of Networks of Low Capability Devices

A Personal View

Giuseppe Persiano

Dipartimento di Informatica ed Appl.
"Renato M. Capocelli"
Università di Salerno

PERADA Summer School – Edinburgh – June 24, 2009



Foundations of Adaptive Networked Societies of Tiny Artefacts

Thanks to:

Vincenzo Auletta, Carlo Blundo, Angelo De Caro, Vincenzo Iovino,
Luca Moscardelli, Paolo Penna, Ivan Visconti.

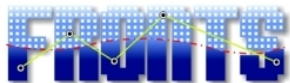
Overview

- A more challenging scenario than Internet security
 - **severely resource-constrained** devices;
 - more serious **privacy threats** than the Internet;
 - new threats deriving from **physical locality**.

Overview

- A more challenging scenario than Internet security
 - **severely resource-constrained** devices;
 - more serious **privacy threats** than the Internet;
 - new threats deriving from **physical locality**.

- Three approaches
 - **Conservative**: scale-down Cryptography
 - **pairings on elliptic curves**;
 - combinatorial cryptography;
 - **Progressive**: new execution framework
 - get help from the environment;
 - **Game Theoretic**: base security of selfishness and rationality;



Outline

The new threats

The conservative approach

The progressive approach

The game theoretic approach

Conclusions

Pervasive Computing Security vs. Internet Security

- **Pervasive Computing:** tens to hundreds of devices per person;
- **Internet:** *a few devices per person:* home pc, workstation, laptop, PDA, smart phone;
- **Pervasive Computing:** users (almost) always on-line and sometimes unaware of that;
- **Internet:** *users on-line in specific time intervals and always aware of connection;*

Pervasive Computing Security vs. Internet Security

- **Pervasive Computing:** used to carry out most every-day activities:
 - taking a bus;
 - entering your office;
 - entering your house;
 -;

- **Internet:** *used for financial and leisure activities: home banking, shopping, video on-demand, on-line videogames,*

- **Pervasive Computing:** active physical environment;
- **Internet:** *no interaction with physical environment;*

The new challenges

- **low** computational power;
 - need for a new framework for security protocols;
 - need to re-design security protocols;
 - new crypto primitives;

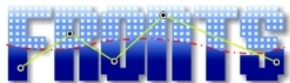
- **myriads**:
 - no centralized control;
 - no a-priori trust structure;
 - mobile device must adapt to environment;

The new challenges

- **low** computational power;
 - need for a new framework for security protocols;
 - need to re-design security protocols;
 - new crypto primitives;

- **myriads**:
 - no centralized control;
 - no a-priori trust structure;
 - mobile device must adapt to environment;

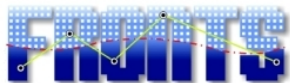
- **spatial localization**:
 - location-based security;
 - location privacy;
 - new attacks are possible (physical attacks);



Low computational power

- Devices **might not** be capable of implementing current cryptographic primitives.

Need to think of different primitives



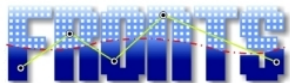
Low computational power

- Devices **might not** be capable of implementing current cryptographic primitives.

Need to think of different primitives

- Devices **are not** be capable of carrying-out current security protocols.

Need to think of different execution frameworks



The conservative approach

Base Security on Cryptography

The conservative approach

Base Security on Cryptography

- **re-founding** Cryptography to take into account the new scenario;
- **re-implementing** Cryptography to make it usable in the new scenario;

Refounding Cryptography

Complexity-based Crypto

One-way functions:

- the user can compute $f(x)$ in polynomial-time;
- the adversary cannot compute $f^{-1}(y)$ in polynomial-time;

Perfect fit for Internet-security: good and bad guys have the same power.

Refounding Cryptography

Complexity-based Crypto

One-way functions:

- the user can compute $f(x)$ in polynomial-time;
- the adversary cannot compute $f^{-1}(y)$ in polynomial-time;

Perfect fit for Internet-security: good and bad guys have the same power.

- Good guy performs his computation in \mathbb{P} ;
- Bad guy performs his computation in \mathbb{P} ;
- Bad guy needs \mathbb{NP} to break security;

Necessary condition: $\mathbb{P} \neq \mathbb{NP}$.

Refounding Cryptography II

Small artifacts

- Good guy is a **DFA**;
- Bad guy performs his computation in \mathbb{P} ;

Refounding Cryptography II

Small artifacts

- Good guy is a **DFA**;
- Bad guy performs his computation in \mathbb{P} ;

One-way functions:

- the user can compute $f(x)$ using a **DFA**;
- the adversary cannot compute $f^{-1}(y)$ in polynomial-time;

Refounding Cryptography II

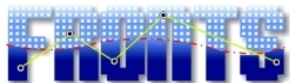
Small artifacts

- Good guy is a **DFA**;
- Bad guy performs his computation in \mathbb{P} ;

One-way functions:

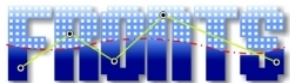
- the user can compute $f(x)$ using a **DFA**;
- the adversary cannot compute $f^{-1}(y)$ in polynomial-time;

Too good to be true (**DFA** = **NFA**).



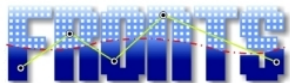
Some good news

- PCP is NP for small devices;



Some good news

- PCP is NP for small devices;
 - a small device can check a proof provided by a more powerful device by checking only 3 bits and only performing XOR operations;



Some good news

- PCP is NP for small devices;
 - a small device can check a proof provided by a more powerful device by checking only 3 bits and only performing XOR operations;
- some cryptography can be done with NC0 overhead;

Some good news

- PCP is NP for small devices;
 - a small device can check a proof provided by a more powerful device by checking only 3 bits and only performing XOR operations;

- some cryptography can be done with NC0 overhead;
 - implementable by shallow circuits;

Some good news

- **PCP is NP for small devices;**
 - a small device can check a proof provided by a more powerful device by checking only 3 bits and only performing XOR operations;

- **some cryptography can be done with NC0 overhead;**
 - implementable by shallow circuits;

- **requires two-way access to input;**
 - cannot implement PRG using log-space one-way automata;

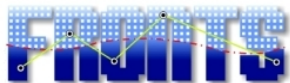
Some good news

- PCP is NP for small devices;
 - a small device can check a proof provided by a more powerful device by checking only 3 bits and only performing XOR operations;

- some cryptography can be done with NC0 overhead;
 - implementable by shallow circuits;

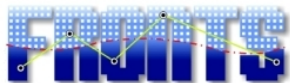
- requires two-way access to input;
 - cannot implement PRG using log-space one-way automata;

... but still a long way to go!



(Long Term) Research Questions

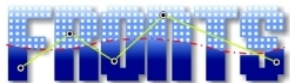
- find the equivalent of one-functions for small devices;
- implement Cryptographic primitives with small overhead;



Foundations of Adaptive Networked Societies of Tiny Artefacts

Re-implementing Cryptography

find more efficient implementations of cryptographic primitives



Re-implementing Cryptography

find more efficient implementations of cryptographic primitives

need help from Number Theorists

Pairing-based Crypto

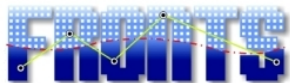
Pairing on Elliptic Curves

(symmetric version)

- small key size and parameters;
- fast group (and crypto) operations;
- low storage and bandwidth;
- multiplicative groups \mathbb{G} and \mathbb{G}_T of **prime** order p ;
- non-degenerate pairing function $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$;
 - for all $x \in \mathbb{G}$, $x \neq 1$, and $a, b \in \mathbb{Z}_p$,

$$e(x, x) \neq 1 \text{ and } e(x^a, x^b) = e(x, x)^{ab}.$$

Examples: Weil and Tate pairings.



Performance Comparison

Elliptic Curves	EIGamal	AES
160 bit	1024 bit	80 bit
256 bit	3072 bit	128 bit
384 bit	8192 bit	192 bit

With pairings:

- good guy is “polynomial” in 160;
- bad guy can be “polynomial” in 1024;
- the system is still secure;

Extra benefits from Pairings

Attribute based encryption

- a public key settings (Pk, MSk) ;
- from MSk can derive a secret key K for every pattern \vec{k} ;
- cyphertexts C have attributes \vec{x} ;
- key K can decrypt ciphertext C iff $P(\vec{x}, \vec{k}) = 1$;

Extra benefits from Pairings

Attribute based encryption

- a public key settings (Pk, MSk);
- from MSk can derive a secret key K for every pattern \vec{k} ;
- cyphertexts C have attributes \vec{x} ;
- key K can decrypt ciphertext C iff $P(\vec{x}, \vec{k}) = 1$;

Efficient implementations can be given for matching with “don’t care” (a.k.a. Hidden Vector Encryption).

Privacy of attributes and pattern can be guaranteed.

Attribute based encryption in a pervasive environment

A scenario

- a large number of artefacts are deployed;
- each has some features (release date, hardware characteristic,...)
- features are encrypted to keep them private;
- artefacts with specific characteristics must be re-called;

Attribute based encryption in a pervasive environment

A scenario

- a large number of artefacts are deployed;
 - each has some features (release date, hardware characteristic,...)
 - features are encrypted to keep them private;
 - artefacts with specific characteristics must be re-called;
-
- release the decryption key to trusted entities;
 - decrypt the information contained in each artefact;
 - select those that match;

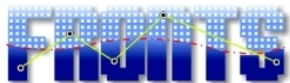
Attribute based encryption in a pervasive environment

Attribute based encryption comes to help

- encrypt attributes;
- release keys corresponding to the desired pattern;

No information is revealed about artefacts:

- only whether they match or not.



Research Questions

- what can we do with pairings for small devices?
- fast alternatives to pairings?

The progressive approach

Getting help from the environment

- computational load for carrying out a protocol can be shared among a group of **security proxies**;
- security proxies can be woven into an **active** environment;
- private information must not be leaked from the device to the proxies;
- proxies are not trusted;
- efficiency;
- no (or very little) infrastructure should be assumed;

Encrypting with the help of proxies

RSA proxies

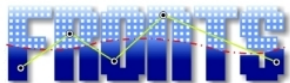
- **Input:** $r, (N, e)$;
- **Output:** $E(r, (N, e)) := r^e \pmod N$.

Using RSA proxies

- 1 Compute $r_1, r_2, r_3 := m \cdot r_1^{-1} r_2^{-1}$.
- 2 Give r_i to proxy i .
- 3 Receive p_i from proxy i .
- 4 Compute $p_1 \cdot p_2 \cdot p_3$.

Cost with proxies: 4 modular multiplications.

Cost without proxies: 1024 modular multiplications.



Encrypting with the help of proxies

Security

Any two r_i 's are independent from m .

Secure if proxies do not talk to each other.

But what if proxies do not compute the RSA function?

Encrypting with the help of proxies

Security

Any two r_i 's are independent from m .
 Secure if proxies do not talk to each other.

But what if proxies do not compute the RSA function?

The trust model

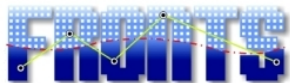
- ① proxies are assumed not to talk to each other;
- ② we rely on proxies to perform the prescribed algorithm;

Radio-Frequency Identification (RFID)

- a microchip that is capable of transmitting a static identifier for a short distance;
- activated by a query from a nearby reader, which also transmits power for the operation of the tag;
- about 3 EuroCent per unit and size about $.4 \times .4$ mm.

RFID Tags have almost no computation power

RFID Tags can be used in conjunction with a reader



Privacy

- RFID will broadcast its content every time it is queried;
- the item to which it is attached (and its owner) can be traced;
- do not want to remove RFID for post-sale management (e.g., return of unwanted items, warranty)
- cannot remove if RFID needed to track borrowed items (e.g., public library)

Randomizing Encryption for RFID Tags

- objects are tagged so that owner can store info on tags;
- info are encrypted so that only owner can decrypt it;
- ciphertexts can be traced;

- **randomizers** are woven into the environment;
- every time a tag is within range
 - the randomizer reads the tag;
 - and re-writes it;
 - without modifying the encrypted content;

Current solutions based on pairings.

Randomizing Encryption for RFID Tags

We have three types of honest players:

- 1 The **Central Authority CA** that publishes some public information Pub and issues a pair of private and secret keys to each authorized player.
- 2 The **tag owners** that receive a public and secret key from the CA and use the keys to encrypt and decrypt messages that are stored on tags.
- 3 The **randomizers** that receive tags and randomize the ciphertexts stored on the tags. The randomization procedure changes the ciphertext but not the cleartext stored on the tag.

The syntax of the scheme

- ① **GenPub** executed by the **CA**; outputs public information Pub and the master secret key Msk.
- ② **GenKey** executed by the **CA**; outputs the secret key SkId of user Id.
- ③ **rE** encrypts a message M to be written on a tag. It takes as inputs Pub, M and the identity Id;
- ④ **rD** decrypts the content of a tag. It takes as inputs Pub and SkId.
- ⑤ **Randomize** executed by the **randomizers**. It takes as inputs Pub and a ciphertext Ct and outputs ciphertext Ct* that encrypts same message M for same user Id. M , Id and the secret key SkId of Id are not needed by Randomize.

skip construction

Hardness Assumptions

Symmetric Decision BDH

Given a tuple $[g, g^{z_1}, g^{z_2}, g^{z_3}, Z]$ for random exponents $z_1, z_2, z_3 \in \mathbb{Z}_p$ it is hard to distinguish between $Z = \mathbf{e}(g, g)^{z_1 z_2 z_3}$ and a random Z random from \mathbb{G}_T .

Symmetric Decision Linear

Given a tuple $[g, g^{z_1}, g^{z_2}, g^{z_1 z_3}, g^{z_2 z_4}, Z]$ for random exponents $z_1, z_2, z_3 \in \mathbb{Z}_p$ it is hard to distinguish between $g^{z_3 + z_4}$ and random Z from \mathbb{G} .

The construction

Procedure GenPub

Pick $t_1, t_2, t_3, \omega, \leftarrow \mathbb{Z}_p$ and $g, g_0, g_1 \leftarrow \mathbb{G}$.

$$\text{Pub} = \begin{bmatrix} g, & g_0, & g_1, & e(g, g)^{\omega t_1 t_2 t_3}, & g^{t_1}, & g^{t_2}, & g^{t_3} \\ g, & g_0, & g_1, & \Omega, & v_1, & v_2, & v_3 \end{bmatrix}.$$

The master secret key is $\text{Msk} = (t_1, t_2, t_3, w)$.

The construction

Procedure GenKey.

Set $\text{Id} = g_0 g_1^{\text{Id}}$, pick $r \leftarrow \mathbb{Z}_p$ and set $\text{SkId} = [D_0, D_1, D_2, D_3]$ as:

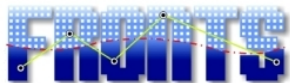
$$\text{SkId} = \begin{bmatrix} g^{rt_1 t_2 t_3}, & g^{-wt_1 t_3} \text{Id}^{-rt_1 t_3}, & g^{-wt_1 t_2} \text{Id}^{-rt_1 t_2}, & g^{-wt_2 t_3} \text{Id}^{-rt_2 t_3}, \\ D_0, & D_1, & D_2, & D_3 \end{bmatrix}.$$

The construction

Basic Encryption $E(\text{Pub}, \text{Id}, M)$

Pick $s, s_1, s_2 \leftarrow \mathbb{Z}_p$ and set

$$E(\text{Pub}, \text{Id}, M) = \begin{bmatrix} \Omega^s \cdot M, & \text{Id}^s, & v_2^{s_2}, & v_3^{s-s_1-s_2}, & v_1^{s_1} \\ C', & C_0, & C_1, & C_2, & C_3 \end{bmatrix}.$$



The construction

Randomizable Encryption $rE(\text{Pub}, \text{Id}, M)$

$$rE(\text{Pub}, \text{Id}, M) = \begin{bmatrix} E(\text{Pub}, \text{Id}, M), & E(\text{Pub}, \text{Id}, 1) \\ C, & U \end{bmatrix}.$$

The construction – Decryption

Let $C = [C', C_0, C_1, C_2, C_3]$ be a ciphertext. Then

$$M = C' \cdot \mathbf{e}(C_0, D_0) \cdot \mathbf{e}(C_1, D_1) \cdot \mathbf{e}(C_2, D_2) \cdot \mathbf{e}(C_3, D_3).$$

Indeed

$$\mathbf{e}(C_0, D_0) \cdot \mathbf{e}(C_1, D_1) \cdot \mathbf{e}(C_2, D_2) \cdot \mathbf{e}(C_3, D_3) = \mathbf{e}(g, g)^{-wt_1 t_2 t_3 s} = \Omega^{-s}.$$

and $C' = M \cdot \Omega^s$.

The construction

Procedure Randomize($C_t = [C, [U', U_0, U_1, U_2, U_3]]$)

Define,

$$U^* = [U'^r, U_0^r, U_1^r v_2^{r_2}, U_2^r v_3^{r_3}, U_3^r v_1^{-r_2-r_3}]$$

for random $r, r_3, r_2 \in \mathbb{Z}_p$.

Return (C^*, U^{**}) where $C^* = C \cdot U^*$ and $U^{**} = (U^*)^*$.

The security model

- Adversary picks two tags T_0 and T_1 and encrypts on each of them a message of his choice for any identity of his choice.

We require that no poly-time adversary can win with probability significantly higher than $1/2$.

Drawback: Adversary can only encrypt.

back

The security model

- Adversary picks two tags T_0 and T_1 and **encrypts** on each of them a message of his choice for any identity of his choice.
- b is chosen at random from $\{0, 1\}$ and the randomized version of the content of T_b is given to the adversary.

We require that no poly-time adversary can win with probability significantly higher than $1/2$.

Drawback: Adversary can only encrypt.

back

The security model

- Adversary picks two tags T_0 and T_1 and **encrypts** on each of them a message of his choice for any identity of his choice.
- b is chosen at random from $\{0, 1\}$ and the randomized version of the content of T_b is given to the adversary.
- **The adversary outputs \tilde{b} and wins if $b = \tilde{b}$.**

We require that no poly-time adversary can win with probability significantly higher than $1/2$.

Drawback: Adversary can only encrypt.

back

A modified construction

Ciphertexts

A ciphertext consists

- a basic encryption of message M ;
- an encryption of identity Id of tag owner using a secret key known to CA and randomizers.

Randomizing

Randomization procedure of $Ct = [C, Z]$ consists of:

- first decrypt Z to get Id ;
- compute basic encryption U of 1 w.r.t. to Id .
- compute new encryption Z' of Id ;
- output $[C \cdot U, Z']$;

Security model

- Randomizers are assumed to be semi-trusted.
- A randomizer knows which tag belongs to which tag-owner.
- Still cannot distinguish between tags of the same user.
 - randomizers know if it is an Armani jacket or a Gucci jacket;
 - still cannot distinguish between two jackets from the same maison;

Notice: introduction of new objects and new tag owners to the system is trivial.

Trust Model for Randomizing RFID

The trust model

- ① randomizers are unable to trace tags;
- ② we rely on randomizers to perform the prescribed algorithm;
 - if they don't still RFID tags are untraceable;
 - content might be destroyed;

Identification Based Trust Establishment

The paradigm

- each player has
 - a pair of public/secret key (P_k, S_k) ;
 - a list of friends' public keys $(P_{k_1}, \dots, P_{k_\ell})$;

- if two players interact
 - receive from the other player his public key;
 - check if on list of friends;
 - if it is, **verify possession of associated private key**;

A concrete scenario

EPassports

- a small device is implanted in a passport;
- at the border a protocol is carried out to assert validity of the passport;
- for privacy reasons the protocol should not give the border control guard a *transferable* proof that the protocol took place.

A concrete scenario

EPassports

- a small device is implanted in a passport;
- at the border a protocol is carried out to assert validity of the passport;
- for privacy reasons the protocol should not give the border control guard a *transferable* proof that the protocol took place.

We can use *zero-knowledge* identification protocols.

A concrete scenario

EPassports

- a small device is implanted in a passport;
- at the border a protocol is carried out to assert validity of the passport;
- for privacy reasons the protocol should not give the border control guard a *transferable* proof that the protocol took place.

We can use *zero-knowledge* identification protocols.

Resettable ZK

Physical capture

during the protocol the passport is physically held by the guard;

- it is known that by disconnecting power or by applying strong electro-magnetical fields devices reset to initial state;
- the same protocol can be executed more than once with the passport using the same random coins.

We can use *resettable zero-knowledge* identification protocols.

rZK requires

- either an unbounded number of rounds;
- or public-key infrastructure;

Not usable for small devices and for E-passports

Non-Transferability

Non-Transferability for Epassport

- Each passenger has a public key Pk and a secret key Sk .
- The public key is signed by issuing authority that is recognized by G .
- Passenger U (using his passport) and border-control guard G execute protocol Π so that U can prove who he is.
 - U convinces G that knows secret associated with Pk .
- Party G should not be able to run protocol Π' with party V and convince V that G has interacted with U .

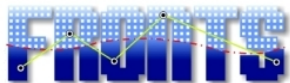
Non-Transferability

Non-Transferability for Epassport

- Each passenger has a public key Pk and a secret key Sk .
- The public key is signed by issuing authority that is recognized by G .
- Passenger U (using his passport) and border-control guard G execute protocol Π so that U can prove who he is.
 - U convinces G that knows secret associated with Pk .
- Party G should not be able to run protocol Π' with party V and convince V that G has interacted with U .

Man-in-the-Middle Attacks

G relays messages between U and V .
 If U convinces G then G convinces V .



Going back to ZK

Obstacles

- challenge-response protocols;
- challenges chosen by the verifier;
- need to be secure for all possible challenges;

Going back to ZK

Obstacles

- challenge-response protocols;
- challenges chosen by the verifier;
- need to be secure for all possible challenges;

Random beacon: a simple security proxy

- introduced by Rabin for Byzantine Agreement;
- at regular intervals, a number of random bits are announced to all players;
- physically implementable and thus no trust model!
- most of the impossibility results for ZK disappear;

Space bounded crypto

Key agreement with RB

- each artifacts picks fraction p at random of the bits coming from RB;
- some bits will be picked from both;
- XOR them together and get 1 bit;
- repeat until you have enough bits;

With high probability, an adversary (even if it has much larger memory) will miss at least one bit that was picked from both parties.

Cryptography vs Game Theory

The Crypto Approach

Authorization is granted based on “*who you are*”.

Major security task: identify who you are, and act accordingly.

Cryptography vs Game Theory

The Crypto Approach

Authorization is granted based on “*who you are*”.

Major security task: identify who you are, and act accordingly.

The Game Theory Approach

Identity is not well defined.

Authorization is granted based on “*how you behave*”.

Major security task: keep track of history.

Goal

Design a protocol for the players:

Design criteria

- the protocol is in equilibrium
 - no player benefits from deviating from the protocol;
- the protocol guarantees a “good” global performance;

Goal

Design a protocol for the players:

Design criteria

- the protocol is in equilibrium
 - no player benefits from deviating from the protocol;
- the protocol guarantees a “good” global performance;

Trust

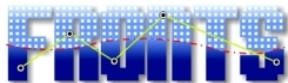
- is based on **selfishness** and **rationality**:
 - it is in my interest to follow the protocol;
- is not based on **friendship**:
 - I prove to you that I am on your list of friends;

A simple game between two players

The scenario

- A and B wish to send a message;
- can choose the power of transmission: 0 (no transmission), 1 (lowest possible power), . . . , k (highest possible power);
- transmitting at power P costs P ;
- successful transmission is worth $v > k$;
- A transmits at P_A and B transmits at P_B ;
 - if $P_A > P_B$ then A succeeds and B doesn't.
 - if $P_A = P_B$ then neither succeeds.
 - if $P_A < P_B$ then B succeeds and A doesn't.
- $U_A(P_A, P_B) = \chi_A \cdot v - P_A$ and $U_B(P_A, P_B) = \chi_B \cdot v - P_B$;

There is no Pure Nash equilibrium.



An example

Example: $v = 3$ and $k = 2$;

U	0	1	2
0	(0,0)	(0,2)	(0,1)
1	(2,0)	(-1,-1)	(-1,1)
2	(1,0)	(1,-1)	(-2,-2)

An example

Example: $v = 3$ and $k = 2$;

U	0	1	2
0	(0,0)	(0,2)	(0,1)
1	(2,0)	(-1,-1)	(-1,1)
2	(1,0)	(1,-1)	(-2,-2)

P	0	1	2
0	0	$2/7$	$1/7$
1	$2/7$	0	$1/14$
2	$1/7$	$1/14$	0

This correlated equilibria has social welfare $10/7$.

An example

Example: $v = 3$ and $k = 2$;

U	0	1	2
0	(0,0)	(0,2)	(0,1)
1	(2,0)	(-1,-1)	(-1,1)
2	(1,0)	(1,-1)	(-2,-2)

P	0	1	2
0	0	$2/7$	$1/7$
1	$2/7$	0	$1/14$
2	$1/7$	$1/14$	0

This correlated equilibria has social welfare $10/7$.

Requires existence of a mediator that (**privately**) sends signals to the two parties.

The trust model of correlated equilibria

Trust model

- mediator assumed to sample the correct distribution;
- mediator assumed not to tell the output given to other party;
- private channels between mediator and each of the party.

A Stackelberg game

At each step, the leader transmits with probability p at highest power.

Example: $v = 3$ and $k = 2$;

U	0	1	2
0	(0,0)	(0,2)	(0,1)
1	(2,0)	(-1,-1)	(-1,1)
2	(1,0)	(1,-1)	(-2,-2)

A Stackelberg game

At each step, the leader transmits with probability p at highest power.

Example: $v = 3$ and $k = 2$;

U	0	1	2
0	(0,0)	(0,2)	(0,1)
1	(2,0)	(-1,-1)	(-1,1)
2	(1,0)	(1,-1)	(-2,-2)

U	0	1	2
0	(0,0)	(0,2-3p)	(0,1-3p)
1	(2-3p,0)	(-1,-1)	(-1,1-3p)
2	(1-3p,0)	(1-3p,-1)	(-2,-2)

Pure Nash equilibria of social cost $2 - 3p$ appear for $p \geq 1/3$.

The trust model of Stackelberg leader

Trust model

- leader assumed to sample the correct distribution;
- leader assumed not to tell the output in advance.

Repeated games

Players want to maximize δ -discounted utility: $\sum_i \delta^i u_i$.

Strategy determines, given the partial history, the next move of the player (i.e., power of next transmission).

Subgame Perfect Equilibrium

A strategy \mathcal{S} is Subgame Perfect Equilibrium if, for any partial history \mathcal{H} , \mathcal{S} selects the move that maximizes the discounted utility.

This holds also partial history \mathcal{H} that will **not** occur if both players follow \mathcal{S} .

Optimal strategy

Folk Theorem for Discounted SPE

- players take turn in transmitting at minimum power;
- if at any time, one player deviates from the protocol the other starts the punishment
 - transmission at maximum power for $L = L(\delta)$ turns;
 - best response for the punished player is **no transmission**;
 - forcing the max min (or **threat point**);

A player that fails to punish the other gets punished himself.

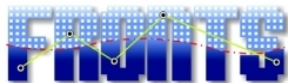
Optimal strategy

Folk Theorem for Discounted SPE

- players take turn in transmitting at minimum power;
- if at any time, one player deviates from the protocol the other starts the punishment
 - transmission at maximum power for $L = L(\delta)$ turns;
 - best response for the punished player is **no transmission**;
 - forcing the max min (or **threat point**);

A player that fails to punish the other gets punished himself.

This strategy guarantees **optimal** social welfare $(v - 1)/2$ for sufficiently large $\delta < 1$.



Full monitoring

The Folk Theorem assumes **full monitoring** each player knows exactly what the previous moves of the other player.

Full monitoring in our game

During transmission the player being punished has to check whether the other player is performing the punishment.

Needs to know the power at which the other player is transmitting.

In reality...

Each player knows all of his previous moves (transmission power) and whether or not he was successful.

Full monitoring

The Folk Theorem assumes **full monitoring** each player knows exactly what the previous moves of the other player.

Full monitoring in our game

During transmission the player being punished has to check whether the other player is performing the punishment.

Needs to know the power at which the other player is transmitting.

In reality...

Each player knows all of his previous moves (transmission power) and whether or not he was successful.

In general **Folk Theorem does not hold** but

Full monitoring

The Folk Theorem assumes **full monitoring** each player knows exactly what the previous moves of the other player.

Full monitoring in our game

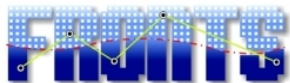
During transmission the player being punished has to check whether the other player is performing the punishment.

Needs to know the power at which the other player is transmitting.

In reality...

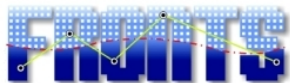
Each player knows all of his previous moves (transmission power) and whether or not he was successful.

In general **Folk Theorem does not hold** but in the specific **we got lucky** (only for two players).



Who designs the algorithm

Algorithm designer has to have (almost) complete knowledge of the scenario.



Who designs the algorithm

Algorithm designer has to have (almost) complete knowledge of the scenario.

Impossible in highly dynamic scenari.

Who designs the algorithm

Algorithm designer has to have (almost) complete knowledge of the scenario.

Impossible in highly dynamic scenari.

Who designs the algorithm

- what happens in repeated games?
- in Nature, **evolution** is the algorithm designer;
- in advanced societies, **culture** (through imitation) is the algorithm designer;

Who designs the algorithm

Algorithm designer has to have (almost) complete knowledge of the scenario.

Impossible in highly dynamic scenari.

Who designs the algorithm

- what happens in repeated games?
- in Nature, **evolution** is the algorithm designer;
- in advanced societies, **culture** (through imitation) is the algorithm designer;
- for small artefacts, **adaptive heuristics** can lead to good equilibria;

Small devices can reach a good equilibrium

Adaptive Heuristics

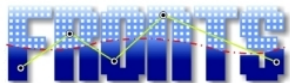
- player is playing strategy k ;
- compute for every other strategy j , **regret**

$$R(j, k) = U(j \rightarrow k) - U(k);$$

- discard strategies with negative regrets;
- choose strategy j with probability \sim to $R(j, k)$;

Theorem (Hart+Mas Collé)

The process converges to a Correlated Equilibrium.



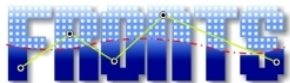
Research question

- find a suitable notion of equilibrium that can be reached by small devices;
- design and analyze scenario in which equilibria with good performance can be reached quickly;

Conclusions

- A more challenging scenario than Internet security
 - **severely resource-constrained** devices;
 - more serious **privacy threats** than the Internet;
 - new threats deriving from **physical locality**.
- Three approaches
 - **Conservative**: scale-down Cryptography
 - **pairings on elliptic curves**;
 - combinatorial cryptography;
 - **Progressive**: new execution framework
 - get help from the environment;
 - **Game Theoretic**: base security of selfishness and rationality;

more on RFID



Foundations of Adaptive Networked Societies of Tiny Artefacts

Thank You!