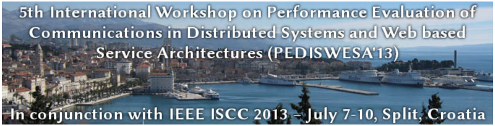
Data protection in multipaths wireless sensor networks

Quentin Monnet[1] Lynda Mokdad[1] Jalel Ben Othman[2]

[1]LACL
Université Paris-Est (FR)

[2]L2TI
Université Paris 13 (FR)



Outline

1. Data protection in WSNs
   - Context
   - Related works
2. Three different methods
   - Global view
   - Securing Data based on Multipaths Routing (SDMP)
   - Shamir's Secret Sharing Scheme (SSSS)
   - Strong encryption
3. Comparison and conclusion
   - Comparison
   - Conclusion
   - Future work

Context

Data protection

Prevent eavesdropping (ensure confidentiality)

in

Wireless Sensor Networks (WSNs)

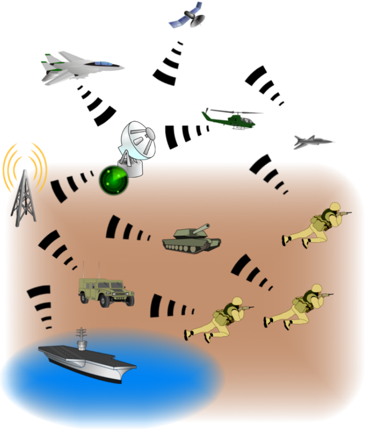
Wireless Sensor Networks (WSNs)

Small devices

- realize measurements (sensors)
- ad-hoc communication
- linked to a base station (BS)

Restricted resources

- few computation capabilities
- few memory available
- few energy available (battery)



Context

Data protection

Prevent eavesdropping (ensure confidentiality)

↘ in

Wireless Sensor Networks (WSNs)

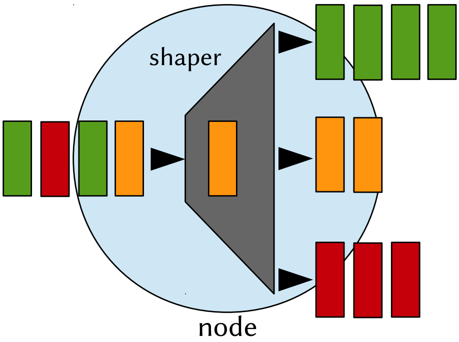
- Low resources
- Strong encryption requires memory and processing time → how to avoid it?

Security in WSNs

Some related works

- **Confidentiality:** A. Babu Karuppiah and S. Rajaram, "Energy efficient encryption algorithm for wireless sensor network", *International Journal of Engineering Research and Technology*, vol. 1, no. 3, May 2012.
- **Multipaths based:** E. Stavrou and A. Pitsillides, "A survey on secure multipath routing protocols in WSNs", *Computer Networks*, vol. 54, no. 13, pp. 2215–2238, Sep. 2010.
- **Other issues** (authentication, DoS, ...): P. Ballarini, L. Mokdad, and Q. Monnet, "Modeling tools for detecting DoS attacks in WSNs", *Security and Communication Networks*, vol. 6, no. 4, pp. 420–436, Apr. 2013.

Classification of packets



Low importance packets	→	Securing Data based on Multi-Path routing
Middle importance packets	→	(Shamir's) Secret Sharing Scheme
High importance packets	→	Strong encryption

Classification is done according to protocols, ports, tags, content, ...

SDMP: Principle

For low importance packets.

1. Split the original into *n* pieces of equal length;
2. Apply XOR (bitwise exclusive "or") operation between fragments;
3. Send each obfuscated fragment through one of the distinct paths of the network;
4. One fragment is sent in clear to enable the receiver to rebuild the message.

SDMP: Splitting the packet

Original message, length = $13 = 4 \times 4 - 3$

Header	72	101	108	108	111	44	32	119	111	114	108	100	33

SDMP: Splitting the packet

Original message, length = 13 = 4 × 4 − 3

Header	72	101	108	108	111	44	32	119	111	114	108	100	33

Randomly padded message, length = 16 = 4 × 4

padding

Header	72	101	108	108	111	44	32	119	111	114	108	100	33	67	12	81

SDMP: Splitting the packet

Original message, length = 13 = 4 × 4 − 3

Header	72	101	108	108	111	44	32	119	111	114	108	100	33

Randomly padded message, length = 16 = 4 × 4

padding

Header	72	101	108	108	111	44	32	119	111	114	108	100	33	67	12	81

$M = p_1 \; p_2 \; p_3 \; p_4$

SDMP: Splitting the packet

Original message, length = 13 = 4 × 4 − 3

Header	72	101	108	108	111	44	32	119	111	114	108	100	33

Randomly padded message, length = 16 = 4 × 4 (padding: 67, 12, 81)

Header	72	101	108	108	111	44	32	119	111	114	108	100	33	67	12	81

$M = p_1 \; p_2 \; p_3 \; p_4$

p_1	Header	01	00	72	101	108	108
p_2	Header	02	00	111	44	32	119
p_3	Header	03	00	111	114	108	100
p_4	Header	04	03	33	67	12	81

SDMP: Splitting the packet

Original message, length = 13 = 4 × 4 − 3

Header	72	101	108	108	111	44	32	119	111	114	108	100	33

Randomly padded message, length = 16 = 4 × 4 (padding: 67, 12, 81)

Header	72	101	108	108	111	44	32	119	111	114	108	100	33	67	12	81

$M = p_1 \; p_2 \; p_3 \; p_4$

p_1	Header	01	00	72	101	108	108	$\oplus p_2$
p_2	Header	02	00	111	44	32	119	
p_3	Header	03	00	111	114	108	100	$\oplus p_4$
p_4	Header	04	03	33	67	12	81	$\oplus p_1$

SDMP: Splitting the packet

Original message, length = 13 = 4 × 4 − 3

Header	72	101	108	108	111	44	32	119	111	114	108	100	33

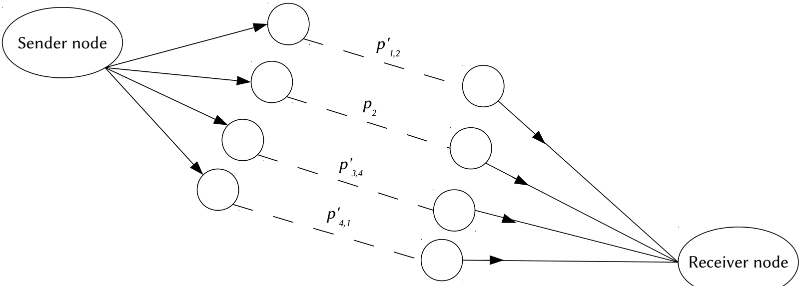
Randomly padded message, length = 16 = 4 × 4 (padding: 67, 12, 81)

Header	72	101	108	108	111	44	32	119	111	114	108	100	33	67	12	81

$M = p_1 \; p_2 \; p_3 \; p_4$

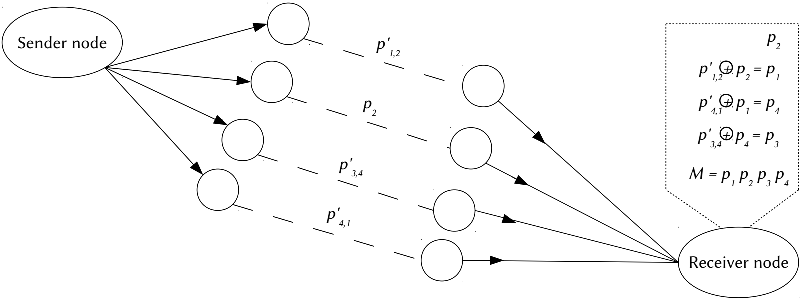
p_1	Header	01	00	72	101	108	108	$\oplus p_2 =$	$p'_{1,2}$	Header	01	00	39	73	76	27
p_2	Header	02	00	111	44	32	119		p_2	Header	02	00	111	44	32	119
p_3	Header	03	00	111	114	108	100	$\oplus p_4 =$	$p'_{3,4}$	Header	03	00	78	49	96	53
p_4	Header	04	03	33	67	12	81	$\oplus p_1 =$	$p'_{4,1}$	Header	04	03	105	38	96	61

SDMP: Sending data



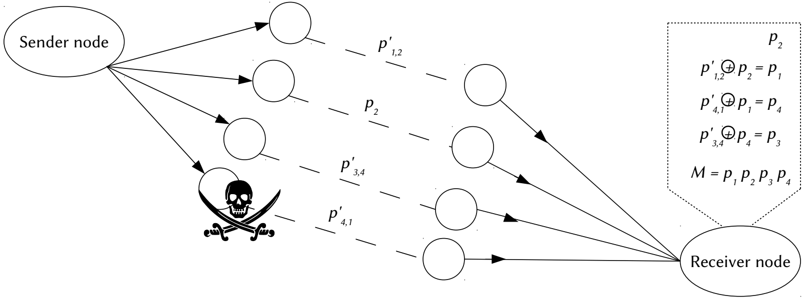
Each chunk of the original message is sent through a different path. The message will be rebuilt by the receiver.

SDMP: Rebuilding the message; signaling



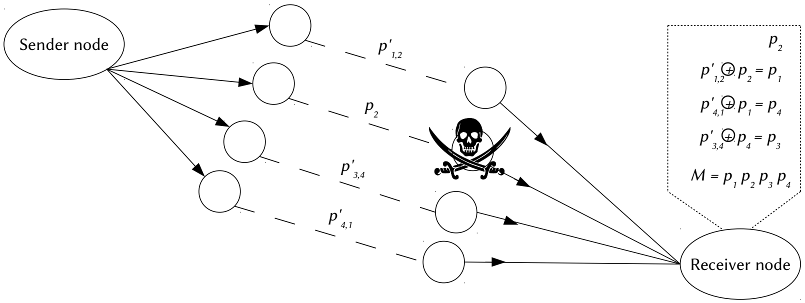
Rebuilding is easy, but receiver needs to know the number of the "key" fragment, as well as the total number of fragments.

SDMP: Attacked!



The attacker is unable to retreive the message without the "key" fragment.

SDMP: Attacked!



The attacker can access clear text, maybe decipher some other fragments!

SDMP: Proposed enhancements

Do not send clear text

Instead of sending the "key" fragment (p_2 in the example) in clear text, send something like $p'_{1,2,3} = p_1 \oplus p_2 \oplus p_3$, then compute $p'_{1,2} \oplus p'_{1,2,3}$ to retreive p_2.

Shuffle the order for the XOR operations

Instead of computing $p'_{1,2}$, $p'_{3,4}$, $p'_{4,5}$, ..., let's compute $p'_{1,4}$, $p'_{3,7}$, $p'_{4,6}$, ...

Secret sharing scheme

For middle importance packets.

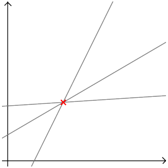
Sharing a secret

- *n* participants sharing a secret
- Any *k* participants are able to recover the secret
- All combinations of $k - 1$ participants must fail to retreive/get information about the secret

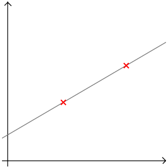
In multipath network

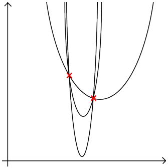
Each one of the *k* part is sent through a distinct path

Shamir's Secret Sharing Scheme: Analogy

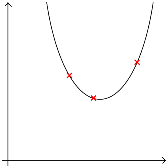


Single point: infinite number of lines.
Two points: one single line.





Same thing for the number of parabolas passing by two distinct points.



Principle: x_1, x_2, ..., x_n are given.
Each participant i knows $y_i = f(x_i)$ (one point).
At least three participants to retreive the equation of a parabola.

Shamir's Secret Sharing Scheme: Principle

The secret is $a_{k-1} \dots a_2 a_1 a_0$
(a_i are elements over the finite field $\mathbb{Z}_p$, p being a prime number)
We consider the associated polynomial function f:

$$f(x) = (a_{k-1}x^{k-1} + \dots + a_2x^2 + a_1x + a_0) \bmod(p)$$

We choose x_1, ..., x_k and compute the secret shares $f(x_1)$, ..., $f(x_k)$.
The original polynomial function, and hence the secret, can be retreived thanks to Lagrange formula:

$$f(x) = \sum_{i=1}^{k} \left(y_i \prod_{\substack{1 \leq j \leq k \\ j \neq i}} \frac{x - x_j}{x_i - x_j} \right)$$

SSSS: Splitting the packet

Original message: 72 101 108 108 111 44 32 119 111 114 108 100 33.

1. We choose $n = 3$ and $k = n$, so $k = 3$; $p = 257$
2. We split the padded message into chunks of length k. 72 101 108, 108 111 32, 44 119 111, 114 108 100 and 33 0 1.
3. We get the five following polynomial functions of degree 2:

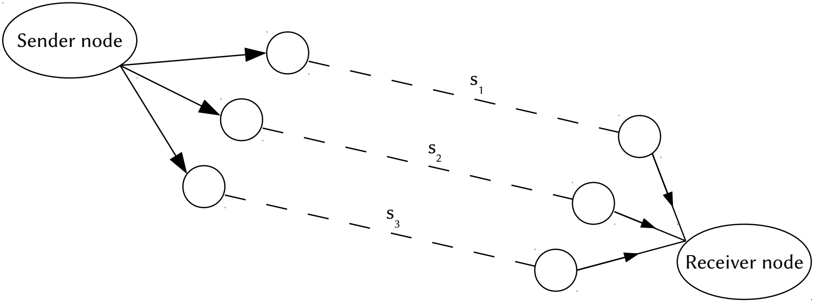
$$\begin{cases} f_1(x) = (72x^2 + 101x + 108) \bmod(257) \\ f_2(x) = (108x^2 + 111x + 32) \bmod(257) \\ f_3(x) = (44x^2 + 119x + 111) \bmod(257) \\ f_4(x) = (114x^2 + 108x + 100) \bmod(257) \\ f_5(x) = (33x^2 + 1) \bmod(257) \end{cases}$$

SSSS: Splitting the packet

1. We choose n distinct and **non-zero** values for x. For instance, $x_1 = 1$, $x_2 = 3$, and $x_3 = 4$.
2. We compute the shares' content.
   The first share (s_1) is $f_1(x_1)$, $f_2(x_1)$, $f_3(x_1)$, $f_4(x_1)$ and $f_5(x_1)$.

Share	Content				
s_1	24	251	17	65	34
s_2	31	52	93	165	41
s_3	122	148	6	43	15

SSSS: Sending data



SSSS: Applying Lagrange formula

$$f_1(x) = \sum_{i=1}^{i=3} \left(f_1(x_i) \prod_{\substack{1 \leq j \leq 3 \\ j \neq i}} \frac{x - x_j}{x_i - x_j} \right)$$

So:

$$f_1(x) = (t_{1,1}(x) + t_{1,2}(x) + t_{1,3}(x)) \bmod(257)$$

where

$$\begin{cases} t_{1,1}(x) = f_1(x_1) \cdot \frac{x - x_2}{x_1 - x_2} \cdot \frac{x - x_3}{x_1 - x_3} = 24 \cdot \frac{x-3}{1-3} \cdot \frac{x-4}{1-4} \\ t_{1,2}(x) = f_1(x_2) \cdot \frac{x - x_1}{x_2 - x_1} \cdot \frac{x - x_3}{x_2 - x_3} = 31 \cdot \frac{x-1}{3-1} \cdot \frac{x-4}{3-4} \\ t_{1,3}(x) = f_1(x_3) \cdot \frac{x - x_1}{x_3 - x_1} \cdot \frac{x - x_2}{x_3 - x_2} = 122 \cdot \frac{x-1}{4-1} \cdot \frac{x-3}{4-3} \end{cases}$$

SSSS: Applying Lagrange formula

$$\begin{cases} t_{1,1}(x) = 24 \cdot \dfrac{(x-3)(x-4)}{255 \cdot 254} = 24 \cdot 6^{-1} \cdot (x-3)(x-4) \\ t_{1,2}(x) = 31 \cdot \dfrac{(x-1)(x-4)}{2 \cdot 256} = 31 \cdot 255^{-1} \cdot (x-1)(x-4) \\ t_{1,3}(x) = 122 \cdot \dfrac{(x-1)(x-3)}{3 \cdot 1} = 122 \cdot 3^{-1} \cdot (x-1)(x-3) \end{cases}$$

All operations are made over the finite field $\mathbb{Z}_p$.

SSSS: Applying Lagrange formula

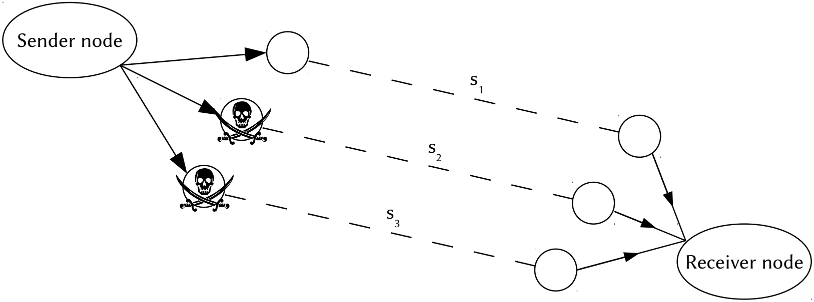
$$\begin{cases} t_{1,1}(x) = 24 \cdot 43 \cdot (x^2 + 250x + 12) = 4 \cdot (x^2 + 250x + 12) \\ t_{1,2}(x) = 31 \cdot 128 \cdot (x^2 + 252x + 4) = 113 \cdot (x^2 + 252x + 4) \\ t_{1,3}(x) = 122 \cdot 86 \cdot (x^2 + 253x + 3) = 212 \cdot (x^2 + 253x + 3) \end{cases}$$

Finally we sum up all $t_{1,i}$ to find:

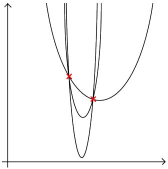
$$f_1(x) = 72x^2 + 101x + 108$$

(Original message: 72 101 108 108 111 44 32 119 111 114 108 100 33)

SSSS: Attacked!



Equivalent to having two points to find a parabola.



Strong encryption

For high importance packets.

Many existing ciphering algorithms

- AES...
- some specific to WSNs

We are not experts in cryptography. We do not propose a new cryptographic algorithm.

(Brief) comparison of the three methods

Method	Confidentiality	Complexity	Overhead
SDMP	Very poor	Very low	Low
SSSS	Low (crypt-analysis?)	$\mathcal{O}(k^2)$, low when k is low	Low
Strong encryption	Very good	High	High, but concerns only one packet

Conclusion

Proposed solution

Traffic shaper to determine importance of packets:

- Low importance → *Securing Data based on Multi-Path routing* method (weak, but fast)
- Middle importance → secret sharing scheme ("medium")
- High importance → strong encryption (much more secure, much heavier)

Also...

- Possible improvements for SMDP
- Detailed example for SDMP and SSSS

Future work

What to do now

- Simulations $\rightarrow$ numerical results, evaluation of performance
- Secret sharing schemes may also be used for availability

The end

Thank you!

Questions?