Garrett Rent
CMSC456
Prof Papamanthou
Feb 5 2021

Vt seems like such a feel-good, modern idea: Allow everyone in a company to know what everyone else in the company is paid. Such pay transparency has many proponents, who argue that it can help motivate employees, close the gender pay gap, and attract new talent by providing a clear picture of how achievement in the company is rewarded. But it can also have the opposite effect, demoralizing employees and driving valuable talent away, especially when it isn't clear why some people are paid more than others. So for organizations considering where to go with pay transparency, it's important to understand why total transparency works in some situations--and can be a disaster in others. Let's dispense with what I consider the easy decisions first. For starters, just about every organization can benefit from some greater level of pay transparency. For instance, if an organization has formal prohibitions on employees sharing salary information with their peers, it's probably time to discard them. Many employees and potential employees find such prohibitions unreasonable, so this is no way to build an effective, motivated workforce. And the legal basis for such rules is at best murky--indeed, a large number of U.S. states now explicitly ban them. The next step in transparency is actively disclosing the processes or formulas used to set pay for all levels of the organization. There is no clear downside to this. For a pay system to be an effective motivational tool, it has to lay out a clear path for employees to earn more by improving their performance. But what about the final step--disclosing everyone's actual level of compensation to all employees? The choice here is more complicated. Full pay transparency works well in two settings. One is where pay levels are based simply on rank and tenure, and perhaps location--not on performance. Think government agencies. On the pay scale for

civilian employees of the federal government, everyone can easily find out what someone in a job at the GS-7 level with three years of experience should earn in a given locale. This lets everyone know exactly what they have to do to earn more, and there is no reason for anyone to feel cheated because someone else is being paid more for doing the same job. Full transparency also appears to work well when everyone in the organization can see how everyone else is performing, especially when performance can easily be measured objectively--for instance, a sales organization that tracks individual sales performance for all to see. Here pay can be based on performance, and the fairness of pay differentials should be clear to all. Another example is small startup firms where all employees are familiar with each other and directly observe each other's performance. It may be difficult to measure performance objectively in these companies, but at least it's reasonably apparent to all what everyone is doing to earn their pay. Move outside those two types of organizations, however, and full transparency becomes a lot trickier. What if a company wants to reward individual performance, but individual contributions are neither visible to everyone in the company nor easy to measure in a fully objective way? What if an organization is relatively large and the work is highly collaborative, making it impossible for everyone to be aware of all their co-workers' contributions and difficult for everyone to agree on how those contributions should be valued? In these settings, full pay transparency can do more harm than good. Excepted from an article by Todd Zenger published in the Wall Street Journal online edition on August 13, 2017.

Code:

```
import string
c = open('ctext.txt', 'r')
ctext = c.read()
def slicen(s, n, truncate=False):
    assert n > 0
    while len(s) >= n:
        yield s[:n]
        s = s[n:]
    if len(s) and not truncate:
        yield s
def originalText(cipher_text):
    key = [186, 31, 145, 178, 83, 205, 62]
    count = 0
    orig_text = []
    for op, code in slicen(cipher_text, 2):
        if op == '0':    op = 0
        elif op == '1':    op = 1
        elif op == '2':    op = 2
        elif op == '3':    op = 3
        elif op == '4':    op = 4
        elif op == '5':    op = 5
        elif op == '6':    op = 6
        elif op == '7':    op = 7
        elif op == '8':    op = 8
        elif op == '9':    op = 9
        elif op == 'A':    op = 10
        elif op == 'B':    op = 11
        elif op == 'C':    op = 12
        elif op == 'D':    op = 13
        elif op == 'E':    op = 14
        elif op == 'F':    op = 15
        else: op = 0
        if code == '0':    code = 0
        elif code == '1':    code = 1
        elif code == '2':    code = 2
```

```
        elif code == '3':    code = 3
        elif code == '4':    code = 4
        elif code == '5':    code = 5
        elif code == '6':    code = 6
        elif code == '7':    code = 7
        elif code == '8':    code = 8
        elif code == '9':    code = 9
        elif code == 'A':    code = 10
        elif code == 'B':    code = 11
        elif code == 'C':    code = 12
        elif code == 'D':    code = 13
        elif code == 'E':    code = 14
        elif code == 'F':    code = 15
        else: code = 0
        hexval = ((op * 16) + code)
        x = int(hexval) ^ key[count % 7]
        orig_text.append(chr(x))
        count += 1
    return("" . join(orig_text))
otext = originalText(ctext)
c.close()
print(otext)
```